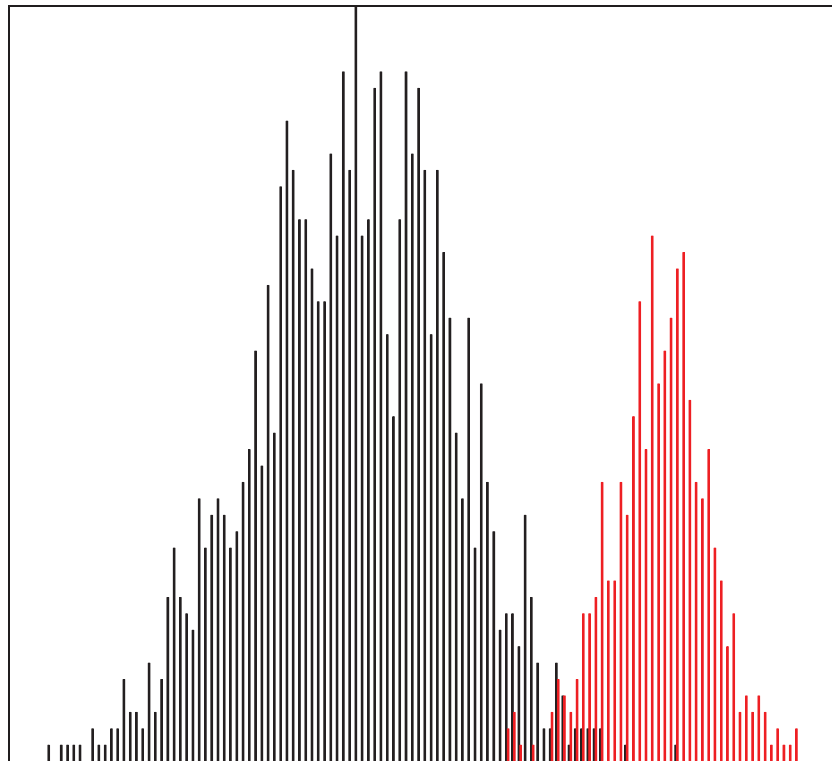


Mehrheitsbildung in der Kombinatorischen Optimierung

Dissertation

zur Erlangung des akademischen Grades
doctor rerum naturalium (Dr. rer. nat.)



vorgelegt dem Rat der Fakultät für Mathematik und Informatik
der Friedrich-Schiller-Universität Jena

von Dipl.-Math. Claus Rose,
geboren am 13.09.1968 in Vilshofen

Jena, September 2000

Claus Rose

Mehrheitsbildung
in der Kombinatorischen Optimierung

Gutachter

1. Prof. Dr. Ingo Althöfer, Jena
2. Prof. Dr. Otto Opitz, Augsburg
3. Prof. Dr. Andreas Dress, Bielefeld

Tag des Rigorosums: 18. Januar 2001

Tag der öffentlichen Verteidigung: 31. Januar 2001

Vorwort

Im Oktober 1996 berichtete mir Ingo Althöfer von seinen Beobachtungen, dass mehrere verschiedene Schachcomputer als gemeinsames „Team“ stärker spielen können als jeder beteiligte Einzelcomputer. Von dem Team galt ein Spielzug als vorgeschlagen, wenn er unter allen Einzelschritten der Computer am häufigsten vorgekommen war. Durch seine Erfolge im Computerschach hinreichend motiviert, setzte Ingo Althöfer als mein Doktorvater mich darauf an, Mehrheitsansätze in der Kombinatorischen Optimierung auszuprobieren. Seine Worte damals waren: „Ich habe da ein gutes Gefühl. Es kommt sicher was dabei heraus. Im Laufe der Zeit wird sich auch ein mathematischer Kern entwickeln.“

Innerhalb von knapp vier Jahren ist eine mathematische Dissertation daraus geworden. Ich möchte Ingo Althöfer an dieser Stelle für seine wegweisende Intuition und die großartige Betreuung während dieser Zeit danken. Wenn sich auch keine tiefliegende mathematische Theorie entwickelt hat, schafft dennoch eine Vielzahl von Experimenten einen umfassenden Eindruck der Wirkungsweise der Mehrheitsbildung bei einer Reihe von Optimierungsproblemen.

Mein besonderer Dank gilt dem Rechenzentrum der Fakultät für Mathematik und Informatik sowie dem Institut für Angewandte Mathematik der Friedrich-Schiller-Universität Jena. Bei der Erstellung dieser Arbeit wurden alle für mich greifbaren Rechner beider Einrichtungen über Jahre exzessiv genutzt. Die verwendeten Computerprogramme wurden ausschließlich in der Programmiersprache C++ verfasst.

Weiterhin möchte ich auch den Kollegen und Studenten danken, die in irgendeiner Weise zum Entstehen dieser Arbeit beigetragen haben. Neben Ingo Althöfer sind Lorenz Hempel, André Henning, Ingolf Koch, Volkmar Liebscher, Dieter Schütze und Stefan Schwarz besonders hervorzuheben. Sie alle haben mir in diversen Diskussionen neue Sichtweisen eröffnet und wichtige Anregungen gegeben. Ferner gebührt Dieter Kaiser und Holger Grabow mein Dank für ihre Soforthilfen bei allen computertechnischen Problemen. Schließlich möchte ich mich bei Ulrike Grunert herzlich bedanken, die den Text mit großer Sorgfalt durchsah und sich eigens für mich einem Selbststudium der neuen Regeln der deutschen Rechtschreibung von 1998 unterzog.

Für mich bleibt die Hoffnung, dass über kurz oder lang verschiedenste Anwendungen in irgendeiner Weise vom Nutzen der Mehrheitsbildung profitieren werden. Insbesondere würde ich mir wünschen, dass die zu Grunde liegenden Ideen an ein möglichst großes Publikum aus diversen Wissenschaftsrichtungen herangetragen werden. So wurde versucht, die Arbeit auch für „Nicht-Eingeweihte“ der Kombinatorischen Optimierung oder der Mathematik verständlich zu halten. An verschiedenen Stellen werden kommentarlos Symbole für grundlegende mathematische Standardgrößen verwendet. Der Leser sei in solchen Fällen auf das Symbolverzeichnis im Anhang verwiesen.

Jena, September 2000

Claus Rose

Inhaltsverzeichnis

Vorwort	3
Einleitung	6
I Grundlagen	11
1.1 Varianten der lokalen Suche	14
1.1.1 Volle lokale Suche	15
1.1.2 Schnelle lokale Suche	16
1.1.3 Zyklische schnelle lokale Suche	17
1.1.4 Deterministische und probabilistische lokale Suche	18
1.1.5 Lokale Suche mit Sidesteps	19
1.2 Untersuchte Optimierungsprobleme	21
1.2.1 Max-SAT	21
1.2.2 Ein Dynamisches Optimierungsproblem	22
1.2.3 Ein Shiftregister-Problem	23
1.2.4 Ein Ising-Spینگlas-Problem	25
1.2.5 Ein Rucksack-Problem	27
1.2.6 Das Counting-Ones-Problem	29
II Einstufige Mehrheitsbildung	30
2.1 Ausgewählte Max-SAT-Benchmarks	30
2.1.1 Die Instanz „Hanoi 5“	30
2.1.2 Weitere Max-SAT-Benchmarks	34
2.2 Erfolgsquoten bei zufällig erzeugten Instanzen	36
2.2.1 Erfolgsquoten für Max-3-SAT	38
2.2.2 Erfolgsquoten für das Dynamische Optimierungsproblem	52
2.2.3 Erfolgsquoten für das Shiftregister-Problem	53
2.2.4 Erfolgsquoten für das Ising-Spینگlas-Problem	58
2.2.5 Erfolgsquoten für das Rucksack-Problem	61
2.2.6 Erfolgsquoten für das Counting-Ones-Problem	63
2.3 Diskussion der Misserfolgstäler	66
2.3.1 Misserfolgstäler bei modifizierten Erfolgskurven	68
2.3.2 Erfolg und Misserfolg bei Pool-Erfolgskurven	70
III Randomisierte Mehrheitsbildungen	72
3.1 Einstimmigkeit, Proporz, Antimehrheit	72
3.2 Das allgemeine Modell randomisierter Mehrheitsbildungen	80
IV Mehrstufige Mehrheitsbildung	86
4.1 Mehrstufige Mehrheitsbildung bei „Hanoi 5“	86
4.2 Wiederholte Mehrheitsbildung als Genetischer Algorithmus	90
4.2.1 Algorithmus 4.2 am Beispiel „Hanoi 5“	94

4.2.2	Populationsgröße, Generationenzahl und Elternzahl	97
4.2.3	Rekombination mit Proporz- und Einstimmigkeitsbildung	100
Anhang	Mittelwertbildung bei einem Scheduling-Benchmark und Varianten	104
A 1	Ergebnisse bei „Whizzkids ’97“	111
A 1.1	Häufigkeitsverteilungen lokaler Minima aus zufälligen Läufen und Konsensläufen	111
A 1.2	Wiederholte Mittelwertbildung als Genetischer Algorithmus	113
A 2	Ergebnisse bei zufällig erzeugten Varianten von „Whizzkids ’97“	116
A 2.1	Häufigkeitsverteilungen lokaler Minima aus zufälligen Läufen und Konsensläufen	117
A 2.2	Erfolgsquoten	119
A 2.3	Optimale Populationsgröße und Elternzahl im Genetischen Algorithmus mit wiederholter Mittelwertbildung	121
	Fazit, Ausblick, offene Fragen	123
	Symbolverzeichnis	129
	Literatur	131
	Index	133

Einleitung

In der Kombinatorischen Optimierung hat man es mit einer Aufgabenstellung der folgenden Art zu tun: Gegeben sei eine endliche Menge V sowie eine Abbildung $f : V \rightarrow \mathbb{R}$. Die Elemente von V heißen *Lösungen*. Gesucht ist eine *global maximale* (bzw. *global minimale*) Lösung, d.h. ein $x^* \in V$ mit größtem (bzw. kleinstem) f -Wert unter allen Elementen von V . Die Bestimmung einer global minimalen Lösung bzgl. f entspricht der Ermittlung einer global maximalen Lösung bzgl. $-f$ und umgekehrt. Man kann sich daher auf einen der beiden Fälle zurückziehen. Im Folgenden werden Maximierungsaufgaben angenommen. Oft gehört solch eine Aufgabe einem schwierigen, etwa NP-vollständigen Maximierungsproblem an und ist innerhalb der verfügbaren Rechenzeit nicht zu bewältigen. Man ist dann gezwungen, Algorithmen zur exakten Lösung der Aufgabe durch eine Heuristik zu ersetzen. Heuristiken liefern im Allgemeinen statt einer global maximalen Lösung x^* nur eine oder mehrere *suboptimale Lösungen*. Diese sind Lösungen mit schlechterem, d.h. kleinerem f -Wert als $f(x^*)$, welche aber vom Benutzer als gut oder wenigstens brauchbar angesehen werden. Ihre Bestimmung sollte in vertretbarer Zeit möglich sein.

Statt der unabhängigen Ermittlung von etwa K solchen „guten“ Lösungen bietet sich folgende Alternative an: Man bestimme mittels einer Heuristik zunächst $L = K - 1$ „gute“ Lösungen $x^1, x^2, \dots, x^L \in V$ unabhängig voneinander. Dann beziehe man diese Lösungen in die Ermittlung einer $(L + 1)$ -ten „guten“ Lösung $x^{L+1} \in V$ ein. Im Fall $V = \{0, 1\}^n$ und L *ungerade* besteht hierfür die Möglichkeit, einen Konsens aus x^1, \dots, x^L im folgenden Sinne zu bilden: Für jedes $i \in \{1, \dots, n\}$ wird als i -te Komponente einer Lösung $x \in \{0, 1\}^n$ der mehrheitlich vorkommende Eintrag der i -ten Komponenten von x^1, \dots, x^L gesetzt. Die Erwartung ist, dass die Heuristik mit Startvektor x zu einer Lösung x^{L+1} führt, die besser als x^1, \dots, x^L ist. Bei einer *geraden* Anzahl L ermittelter Lösungen kann es passieren, dass in den i -ten Komponenten für irgendein festes i so viele Nullen wie Einsen auftreten. Dann ist die i -te Komponente von x nicht eindeutig. Die Mehrdeutigkeit kann aber durch zufällige Wahl aus $\{0, 1\}$ „behoben“ werden. Zu Lösungen x^1, \dots, x^L für *beliebiges* L ist also ein Vektor x mit folgender Eigenschaft konstruierbar: *Falls* für festes $i \in \{1, \dots, n\}$ in den i -ten Komponenten von x^1, \dots, x^L eine Mehrheit für 0 oder 1 vorliegt, so steht der mehrheitlich vorkommende Eintrag auch in der i -ten Komponente von x . x heißt deshalb ein *Mehrheitsvektor* zu x^1, \dots, x^L . Die Konstruktion von x heißt *Mehrheitsbildung*. Für die vorliegende Dissertation sind die Mehrheitsbildung im $\{0, 1\}^n$ und ihr Nutzen für die Kombinatorische Optimierung der zentrale Gegenstand aller Untersuchungen.

Außerhalb der Kombinatorischen Optimierung ist man in verschiedensten Situationen mit dem Prinzip, Mehrheitsentscheidungen zu treffen, vertraut. Man denke etwa an diverse Wahlverfahren von demokratischen Parlamenten bis hin zu Vereinsvorständen. Im Bereich des Computerschachs verglich ALTHÖFER [2] die Spielstärke einer Gruppe von Schachcomputern mit der Spielstärke jedes beteiligten Einzelcomputers. Zur Verfügung standen insgesamt 14 verschiedene Computer. Ein Spielzug galt von einer Gruppe von Computern als vorgeschlagen, wenn er unter allen Einzelvorschlägen am häufigsten auftrat. Bei Unentschiedenheit wurde gelöst. Als Maß für die Spielstärke wurde in 13 nachgespielten Weltmeisterpartien die Anzahl derjenigen vorgeschlagenen Züge zu Grunde gelegt, die in den Originalpartien vom jeweiligen Sieger tatsächlich durchgeführt worden waren. Es stell-

te sich heraus, dass die durchschnittliche Spielstärke aller möglichen Gruppen fester Größe $s \in \{1\} \cup \{3, 4, \dots, 14\}$ monoton mit s wächst. Ferner war die beste Gruppe fester Größe $s \in \{3, 4, \dots, 14\}$ für jedes s besser als jeder beteiligte Einzelcomputer. ALTHÖFER [2] und ARLT [4] beobachteten darüber hinaus, dass die Spielstärke einer Gruppe von Schachcomputern noch weiter gesteigert werden kann, wenn der Einfluss jedes Einzelcomputers auf den gemeinsam vorgeschlagenen Zug auf geeignete Weise gewichtet wird.

Man rechnet leicht nach, dass die Mehrheitsvektoren zu L Lösungen $x^1, \dots, x^L \in \{0, 1\}^n$ gerade die Mediane zu x^1, \dots, x^L im diskreten Hyperwürfel $\{0, 1\}^n$ mit der HAMMING-Nachbarschaft sind (Bemerkung 2.12, S. 67). Ist $G = (V, E)$ ein beliebiger ungerichteter, endlicher Graph und sind $x^1, \dots, x^L \in V$, so heißt $x \in V$ *Median* zu $x^1, \dots, x^L \in V$, wenn

$$\sum_{\ell=1}^L d_G(x^\ell, x) = \min_{v \in V} \sum_{\ell=1}^L d_G(x^\ell, v)$$

gilt. $d_G(u, v)$ gibt dabei die Länge eines kürzesten Pfads, d.h. die Zahl seiner Kanten, zwischen zwei Knoten u und v in G an. Mehrheitsbildung im $\{0, 1\}^n$ ist also ein Spezialfall von Medianbildung. Für sehr einfache Graphen G , nämlich bei endlichen Pfaden bzw. Kreisen, untersuchten LIEBSCHER und WALCHER [28] theoretisch das Grenzwertverhalten von iterierter Medianbildung: Gegeben sei eine Wahrscheinlichkeitsverteilung $P^0 = (p_v^0)_{v \in V}$ auf V . Für $t > 0$ sei $P^t = (p_v^t)_{v \in V}$ wie folgt definiert: p_v^t ist für $v \in V$ die Wahrscheinlichkeit, dass v Median zu zwei zufällig gemäß P^{t-1} gewählten Knoten $x, y \in V$ mit $|d_G(v, x) - d_G(v, y)| \leq 1$ ist. Zur Untersuchung standen Konvergenzeigenschaften der Folge $(P^t)_{t \geq 0}$.

Der heutige Stand der Theorie zur Medianbildung ist weit davon entfernt, für komplexere Graphen wie den Hyperwürfel $\{0, 1\}^n$ quantifizierbare Aussagen zum Konvergenzverhalten iterierter Medianbildung zu machen. Ferner sind bei Heuristiken zu praxisrelevanten Optimierungsproblemen in aller Regel nur sehr wenige beobachtete Effekte theoretisch fundierbar. Die große Kluft zwischen den theoretischen Grundlagen und den praktischen Fragestellungen rechtfertigt einen experimentellen Zugang zu dieser Thematik. Im Rahmen der vorliegenden Arbeit werden Beobachtungen zur Mehrheitsbildung bei diversen Optimierungsproblemen mit Lösungsraum $\{0, 1\}^n$ vorgestellt. Im Einzelnen werden Max-SAT, ein Dynamisches Optimierungsproblem, ein Ising-Spinglas-Problem, ein Shiftregister-Problem und ein Rucksack-Problem untersucht. Als Heuristiken werden verschiedene Varianten *lokaler Suche*¹ eingesetzt. Ein Lauf lokaler Suche mit zufällig, in $\{0, 1\}^n$ gleichverteilt gewählter Startlösung wird als *zufälliger Lauf* bezeichnet. Startet lokale Suche dagegen in einem Mehrheitsvektor, soll von einem *Mehrheitslauf* gesprochen werden.

Die lokale Suche wird hinsichtlich ihres Eingabe-Ausgabe-Verhaltens als Black Box aufgefasst. Bei Input einer vorgegebenen Lösung x wird irgendeine „gute“ Lösung y erzeugt. Diese Black Box wird auf L unabhängige, zufällige Startlösungen s^1, s^2, \dots, s^L angewendet. Zu den Ergebnissen x^1, x^2, \dots, x^L bildet man dann einen Mehrheitsvektor x und übergibt diesen als Eingabe an die Black Box. Als Ausgabe wird ein bestimmtes Ergebnis x^{L+1} geliefert. Das gesamte Experiment wiederholt man etliche Male. Dabei werden die f -Werte

¹siehe auch [1]

von x^1, \dots, x^L getrennt vom f -Wert von x^{L+1} gesammelt. Am Ende werden die beiden „Typen“ von f -Werten gegenüber gestellt. Bei einer Reihe von Benchmark-Problemen zeigt sich, dass die f -Werte nach der Mehrheitsbildung typischerweise besser sind als die f -Werte bei zufälliger Startlösung (Abschnitt 2.1).

Neben praxisnahen Benchmarks werden auch zufällig erzeugte Probleminstanzen für die Anwendung der Mehrheitsbildung herangezogen (Abschnitt 2.2). Bei allen untersuchten Optimierungsproblemen im $\{0, 1\}^n$ stellt sich heraus, dass die Zielfunktionswerte aus Mehrheitsläufen *im Durchschnitt* besser sind als die Werte aus zufälligen Läufen. Ferner machen Mehrheitsläufe im Mittel weniger Schritte als zufällige Läufe. Diese Beobachtungen lassen sich für jede Dimension n machen.

Anders fallen die Ergebnisse aus, wenn der „Erfolg“ der Mehrheitsbildung auf folgende Weise gemessen wird: Sei \max_{zf} der beste f -Wert aus $L + 1$ zufälligen Läufen und \max_{Mh} der beste f -Wert aus L zufälligen Läufen und einem anschließenden Mehrheitslauf. Die Startlösung des Mehrheitslaufs sei dabei ein Mehrheitsvektor zu den L Lösungen der vorangegangenen zufälligen Läufe. Bei Vergleich von \max_{zf} und \max_{Mh} sind prinzipiell drei verschiedene Ergebnisse möglich, nämlich $\max_{Mh} > \max_{zf}$, $\max_{Mh} = \max_{zf}$ und $\max_{Mh} < \max_{zf}$. Bei Max-3-SAT, dem Dynamischen Optimierungsproblem, dem Ising-Spindglas-Problem und dem Shiftregister-Problem zeigt sich, dass die Heuristik besonders bei großen Problemparametern n von der Mehrheitsbildung profitiert: Für große n tritt typischerweise $\max_{Mh} > \max_{zf}$ als Ergebnis auf. Seine relative Häufigkeit geht für $n \rightarrow \infty$ gegen 100 %. Bei kleinen Parametern n überwiegen dagegen die Ergebnisse $\max_{Mh} = \max_{zf}$ bzw. $\max_{Mh} < \max_{zf}$. Letzterer Parameterbereich wird als *Misserfolgstal* bezeichnet. Für zufällig erzeugte Instanzen des Rucksack-Problems stellt sich heraus, dass Mehrheitsläufe nur in einem beschränkten Intervall von Parametern n empfehlenswert sind. Das Rucksack-Problem ist das einzige untersuchte Optimierungsproblem, bei dem die Mehrheitsbildung für große Parameter n keinen „Erfolg“ im Sinne einer Überlegenheit von \max_{Mh} gegenüber \max_{zf} aufweist.

Zur besseren Einschätzung der Mehrheitsbildung wird ein allgemeines Modell *randomisierter Mehrheitsbildungen* zur Erzeugung von Startlösungen $x \in \{0, 1\}^n$ aus L Ausgangslösungen $x^1, x^2, \dots, x^L \in \{0, 1\}^n$ aufgestellt. Die i -te Komponente von x wird dabei mit einer gewissen Wahrscheinlichkeit $p_{j_i} \in [0, 1]$ als 1 und mit Rest-Wahrscheinlichkeit $1 - p_{j_i}$ als 0 gewählt. j_i ist dabei die Anzahl der Einsen in den i -ten Komponenten von x^1, \dots, x^L . Sei speziell $L = 3$. Dann gilt im Beispiel der Mehrheitsbildung $p_0 = p_1 = 0$ und $p_2 = p_3 = 1$. Setzt man im allgemeinen Modell aus Symmetriegründen $p_2 = 1 - p_1$ und $p_3 = 1 - p_0$, so hängt die Bildungsregel nur noch von den beiden Größen p_0 und p_1 ab. Sie wird dann als (p_0, p_1) -Regel bezeichnet. Bei zufällig erzeugten Instanzen zu Max-3-SAT treten folgende Effekte auf (Abschnitt 3.2):

1. Unter allen (p_0, p_1) -Regeln mit $(p_0, p_1) \in [0, 1]^2$ führt die Mehrheitsregel typischerweise sowohl zu den besten durchschnittlichen f -Werten als auch zu den kleinsten durchschnittlichen Schrittzahlen. Dies gilt *für alle* getesteten Problemgrößen n .
2. Sei $\max_{(p_0, p_1)}$ der beste f -Wert aus drei zufälligen Läufen und einem vierten Lauf, dessen Startlösung durch Anwendung der (p_0, p_1) -Regel aus den zuvor erzeugten „guten“ Lösungen hervorgeht.

- (a) Für Parameter n *außerhalb* des Misserfolgstals ist die $(0, 0)$ -Regel, d.h. die Mehrheitsregel, empfehlenswert. Es gilt:

$$\max_{(0,0)} \approx \max_{(p_0, p_1) \in [0,1]^2} \max_{(p_0, p_1)}$$

- (b) Für *kleine* Parameter n *innerhalb* des Misserfolgstals führt die *Antimehrheitsregel*, d.h. die $(1, 1)$ -Regel, typischerweise zu den besten Werten $\max_{(p_0, p_1)}$ unter allen $(p_0, p_1) \in [0, 1]^2$. An die drei zufälligen Läufen sollte sich also ein *Antimehrheitslauf* anschließen.
- (c) Für *große* Parameter n *innerhalb* des Misserfolgstals ist typischerweise die $(\frac{1}{2}, \frac{1}{2})$ -Regel optimal. Hier sind also vier zufällige Läufe ratsam.

Wie bereits bemerkt, sind im Fall des Rucksack-Problems Mehrheitsläufe bei zufällig erzeugten Instanzen nur in einem beschränkten Intervall von Problemparametern n empfehlenswert (Abschnitt 2.2.5). Außerhalb dieses Intervalls sind Antimehrheitsläufe den Mehrheitsläufen vorzuziehen (Abschnitt 3.1).

Ein Vergleich der zu erwartenden Werte für $\max_{(p_0, p_1)}$ unter verschiedenen Paaren (p_0, p_1) ist in praktischen Situationen mit *Echtzeit-Bedingungen* von Bedeutung, wenn dem Rechensystem innerhalb kürzester Zeit eine möglichst gute Lösung abgefordert wird. Angenommen, es sind bereits L zufällige Läufe lokaler Suche gemacht worden. Die zur Verfügung stehende Restzeit erlaube nicht mehr als einen weiteren Lauf. Dann steht eine Entscheidung an, nach welcher (p_0, p_1) -Regel die Startlösung für den abschließenden Lauf gebildet werden soll.

Sind die Rechenzeit-Vorgaben großzügig, besteht die Möglichkeit, (randomisierte) Mehrheitsbildung wiederholt anzuwenden. Sie kann beispielsweise als Rekombinationsoperator in Genetischer Suche eingesetzt werden. Bei Max-SAT, dem Shiftregister-Problem oder dem Ising-Spinglas-Problem tendieren die gefundenen f -Werte typischerweise dazu, mit wachsender Generationenzahl besser zu werden. Die Schrittzahlen werden dabei kleiner. Diese Effekte treten bereits ohne Zusatzmaßnahmen wie Elitismus, Mutation oder fitness-abhängige Selektion² auf (Abschnitt 4.2.1).

Bei zufällig gewählten Instanzen des Shiftregister-Problems sowie des Dynamischen Optimierungsproblems werden ferner Ergebnisse zur Optimierung von Populationsgröße ϱ und Elternzahl L in Genetischer Suche vorgestellt (Abschnitt 4.2.2). Die Aussagen sind beim Shiftregister-Problem im Mittel über alle Instanzen zum Parameter $n = 1000$ zu verstehen, beim Dynamischen Optimierungsproblem zum Parameter $n = 10\,000$. Es zeigt sich, dass bei jeder festen Populationsgröße ϱ Elternzahl 2 für die Rekombination mit Mehrheitsbildung optimal ist. Das Optimalitätskriterium ist dabei der beste f -Wert $f^*(\varrho, L)$, der in den Populationen aus 100 Generationen insgesamt auftritt. Bei fester Elternzahl L werden die Werte $f^*(\varrho, L)$ mit wachsender Populationsgröße ϱ besser. Bei der *Einstimmigkeitsbildung*, welche die Startlösung eines Laufs lokaler Suche gemäß der $(0, \frac{1}{2})$ -Regel festlegt, ist Elternzahl 3 oder auch höher (bei kleinen Populationen) ratsam. Die Einstim-

²Eine ausführliche Diskussion zu Elitismus, Mutation und Selektion findet man beispielsweise in [20], [29].

migkeitsbildung mit Elternzahl 3 erbringt typischerweise bessere beste gefundene f -Werte als die Mehrheitsbildung mit beliebiger Elternzahl.

Die vorliegende Dissertation ist in vier Kapitel unterteilt. In Kapitel I werden die verwendeten Varianten lokaler Suche sowie die untersuchten Problemklassen detailliert vorgestellt. Kapitel II beschreibt im Wesentlichen die experimentellen Ergebnisse zu den Vergleichen von \max_{zf} und \max_{Mh} . In Kapitel III wird das Modell der randomisierten Mehrheitsbildungen diskutiert. Kapitel IV beschäftigt sich schließlich mit wiederholter (randomisierter) Mehrheitsbildung, insbesondere in Genetischer Suche.

Im Anhang wird ergänzend ein spezieller Typ von Scheduling-Problemen³ (Stundenplan-Problemen) behandelt. Der Lösungsraum ist hier nicht $\{0, 1\}^n$, sondern eine komplexere Menge V zur Modellierung der Stundenpläne. In einem Stundenplan sind die Anfangs- und Endzeiten aller Termine vermerkt. Ein Konsens aus L Stundenplänen $S^1, S^2, \dots, S^L \in V$ kann wie folgt gebildet werden: Man berechne für jeden Termin den Mittelwert seiner L Anfangszeiten. Dann sortiere man die Mittelwerte der Größe nach aufsteigend. Gemäß dieser Reihenfolge füge man die Termine in einen neuen Stundenplan S ein. Diese *Mittelwertbildung* in V wird analog zur Mehrheitsbildung in $\{0, 1\}^n$ behandelt. Als Heuristik zur Bestimmung „guter“ Stundenpläne wird wieder lokale Suche eingesetzt. Startet lokale Suche in einem Stundenplan, der durch Mittelwertbildung entstanden ist, wird von einem *Konsenslauf* gesprochen.

Bei der Mittelwertbildung treten im Wesentlichen die gleichen Ergebnisse wie bei der Mehrheitsbildung auf: Ab einer bestimmten Problemgröße ist es besser, statt $L + 1$ zufällige Läufe lokaler Suche nur L zufällige Läufe und einen anschließenden Konsenslauf zu machen. Bei kleinen Problemparametern ist dagegen von Konsensläufen abzuraten. Es tritt also auch bei der Mittelwertbildung ein Misserfolg auf (Anhang A 2.2). Ferner sind auch bei Einsatz der Mittelwertbildung zur Rekombination in Genetischer Suche die Beobachtungen mit denen der Mehrheitsbildung vergleichbar. Die ermittelten Zielfunktionswerte tendieren dazu, mit wachsender Generationenzahl besser zu werden. Dies geschieht bei gleichzeitiger Abnahme der durchschnittlichen Schrittzahlen der Läufe lokaler Suche. Beide Tendenzen zeichnen sich ohne Elitismus, Mutation oder fitnessabhängige Selektion ab (Anhang A 1.2).

³siehe auch [8]

Kapitel I

Grundlagen

Der technische Fortschritt innerhalb der letzten fünfzig Jahre hat dem Menschen sowohl im beruflichen als auch im privaten Alltag neue Perspektiven eröffnet. Insbesondere der Einsatz von Computern ist heute in praktisch allen Bereichen des täglichen Lebens kaum noch weg zu denken. Mit den gewachsenen Möglichkeiten sind aber gleichzeitig auch die Ansprüche der Menschen in materieller wie auch in ideeller Hinsicht gestiegen. Wo man sich noch vor wenigen Jahrzehnten mit der bloßen Automatisierung einfacher Handgriffe zufrieden gab, sind heute Hochleistungsrechner selbstverständlich, die die menschliche Arbeitskraft sowohl hinsichtlich ihrer Effektivität als auch ihrer Effizienz um ein Vielfaches übertreffen. Man ist längst dazu übergegangen, komplizierte Probleme vollständig vom Rechner lösen zu lassen, ohne dass menschliche Eingriffe erforderlich sind. Natürlich sind auch Szenarien denkbar, in denen der Mensch interaktiv den Verlauf von Berechnungen beeinflusst und dabei sein eventuelles Expertenwissen einbringt. Interaktive oder halbautomatische Verfahren werden in dieser Arbeit allerdings keine Rolle spielen.

In der Kombinatorischen Optimierung geht man von einer Aufgabenstellung des folgenden Typs aus: Gegeben sei eine endliche Menge V sowie eine Abbildung $f : V \rightarrow \mathbb{R}$, die sogenannte **Zielfunktion**. V wird als **Lösungsraum** bezeichnet, seine Elemente als **Lösungen**. Gesucht ist eine **global maximale Lösung**, d.h. eine Lösung $v \in V$ mit dem größten f -Wert unter allen Elementen von V . Im Prinzip lässt sich diese **Maximierungsaufgabe** in endlich vielen Schritten lösen: Man enumeriert die Elemente von V , bestimmt ihre f -Werte und merkt sich ein oder besser gleich alle Elemente mit maximalem Wert. Ist der Mensch allein mit dieser Aufgabe überfordert, kann er sie an eine Maschine delegieren. Natürlich ist hier eine Formulierung der Zielfunktion f vorauszusetzen, die die Werte $f(v)$ bei Eingabe von v an einen Rechner für diesen evaluierbar macht. Ist V jedoch so groß, dass die zur Verfügung stehende Rechenzeit nicht genügt, scheidet diese Methode aus. Man muss dann dazu übergehen, sich auf die Auswertung einiger weniger Elemente von V zu beschränken.

Trotz der rasanten Entwicklung in der Computerindustrie sind die Grenzen des Rechners bei praxisrelevanten Aufgabenstellungen auch heute in der Regel schnell erreicht. Zwar machen Maschinen mit schnelleren Prozessoren und größerer Speicherkapazität die Lösung komplexerer Aufgaben möglich – die Steigerung der Leistungsfähigkeit steht aber kontinuierlich im Wettlauf mit den menschlichen Ansprüchen an die Rechentechnik.

Ein maßgeblicher Anteil am Fortschritt des Machbaren ist auf die laufende Weiterentwicklung effizienter Algorithmen zurückzuführen. Ihr Entwurf setzt sowohl mathematische Theorie als auch praktische Erfahrung in ständig wachsendem Maß voraus. Meist ist die zu betrachtende Menge V kein völlig amorphes Gebilde. Sie besitzt vielmehr irgendwelche Struktur, die es im Idealfall dem Aufgabensteller ermöglicht, gezielt Algorithmen zu entwerfen, welche einen Großteil schlechter Lösungskandidaten außer Acht lassen. Die

Entscheidung, welche Kandidaten ausscheiden, lässt sich manchmal a priori treffen, oft aber auch erst zu Laufzeiten eines entsprechenden Programms. Ungeachtet dessen bekommt der Benutzer am Ende eine optimale Lösung geliefert.

Bei praxisrelevanten Aufgaben reicht jedoch das Strukturwissen typischerweise nicht dazu aus, den Lösungsraum V hinreichend einzuschränken. Häufig kann man sogar mathematisch beweisen, dass die vorliegende Optimierungsaufgabe Instanz eines praktisch nicht handhabbaren Problems ist. Darunter fallen insbesondere NP-schwere bzw. NP-vollständige Probleme, die sich vermutlich nicht in einer in der Zahl der instanzkodierenden Bits polynomiellen Anzahl an Operationen lösen lassen.¹ In solchen Fällen bleibt dem Problemsteller nur noch übrig, seine Anforderungen abzuspecken. Statt mit einer optimalen Lösung $v \in V$ hat er sich mit einer suboptimalen Lösung zufrieden zu geben, d.h. mit einem Element von V , das den f -Wert zwar nicht maximiert, aber dennoch für den Benutzer brauchbar ist. Für zahlreiche Probleme mit handhabbaren Strukturen sind heute effiziente Approximationsalgorithmen bekannt, deren Ergebnisse vergleichsweise geringfügig am größtmöglichen Wert vorbeigehen.

Die Struktur mancher Probleme ist so schwer algorithmisch fassbar oder ausnutzbar, dass weder effiziente exakte, konstruktive Lösungsalgorithmen noch effiziente gute Approximationsalgorithmen bekannt sind. Im schlimmsten Fall bleibt nur zufälliges oder sogar willkürliches Suchen (*Random Search*): Man greift stichprobenartig Elemente aus V heraus, so lange Rechenzeit übrig ist, und übernimmt am Ende die beste gefundene Lösung. Oft erweist sich die Realität jedoch als nicht ganz so katastrophal. Denkbar ist beispielsweise, dass V eine Struktur enthält, so dass alle oder wenigstens viele gute Lösungen Gemeinsamkeiten aufweisen. Welche Lösungen dabei als gut bezeichnet werden sollten, bleibt noch zu klären.

Betrachte beispielsweise das Problem Max-SAT. In einer Instanz von Max-SAT ist der Lösungsraum V der diskrete Hyperwürfel $\{0, 1\}^n$ und die Zielfunktion $f : V \rightarrow \mathbb{R}$ definiert durch

$$f(x) := \sum_{j=1}^m \max \{L(x) \mid L \in \mathcal{L}_j\},$$

wobei \mathcal{L}_j für jedes $j \in \{1, \dots, m\}$ eine endliche Menge von Abbildungen der Form

$$\begin{aligned} \pi_i : \{0, 1\}^n &\rightarrow \{0, 1\}, & \pi_i(x_1, \dots, x_n) &:= x_i, & i &\in \{1, \dots, n\}, \\ \varrho_i : \{0, 1\}^n &\rightarrow \{0, 1\}, & \varrho_i(x_1, \dots, x_n) &:= 1 - x_i, & i &\in \{1, \dots, n\}, \end{aligned}$$

ist. Falls nun π_i mit $i \in \{1, \dots, n\}$ in mindestens einer Menge \mathcal{L}_j vorkommt, nicht aber ϱ_i , so ist in einer guten Lösung $x = (x_1, \dots, x_n) \in V$ die Komponente $x_i = 1$ zu erwarten. Zumindest verkleinert sich der f -Wert einer beliebigen Lösung nicht, wenn ihre i -te Komponente auf 1 gesetzt wird. In der Gesamtheit der guten Lösungen wird der i -te Eintrag also überwiegend 1 sein. Entsprechend stehen in der i -ten Komponente guter Lösungen mehrheitlich Nullen, wenn nur ϱ_i , nicht aber π_i in $\mathcal{L}_1 \cup \dots \cup \mathcal{L}_m$ vorkommt. Treten die Abbildungen π_i und ϱ_i bei festem i zwar gemischt, aber sehr ungleich verteilt auf, so kann man wenigstens darauf hoffen, dass in guten Lösungen der Eintrag gemäß des mehrheitlich vorkommenden Typs überwiegt.

¹Eine Sammlung NP-vollständiger Probleme findet man in [15].

Sei nun umgekehrt bereits eine Liste guter Lösungen auf irgendeine Weise ermittelt worden. Dann besteht die Hoffnung darin, dass ihre „tragenden Eigenschaften“ in den „Mehrheitsvektor“ einfließen, der durch komponentenweisen Mehrheitsentscheid aus den Ausgangslösungen konstruiert wird. Sei etwa $n = 4$, und seien

$$\begin{aligned}x^1 &= (0, 1, 0, 0), \\x^2 &= (1, 1, 1, 0), \\x^3 &= (0, 1, 1, 1).\end{aligned}$$

In den ersten bzw. vierten Komponenten von x^1, x^2, x^3 kommen mehrheitlich Nullen vor, nämlich jeweils zwei Nullen und nur eine Eins. Die zweiten bzw. dritten Komponenten enthalten dagegen jeweils mehr Einsen als Nullen. Als „Mehrheitsvektor“ ergibt sich deshalb $x = (0, 1, 1, 0)$. Allgemein lässt sich zu je L Lösungen $x^1, x^2, \dots, x^L \in \{0, 1\}^n$ ein eindeutiger Vektor $x \in \{0, 1\}^n$ durch komponentenweisen Mehrheitsentscheid konstruieren, falls L *ungerade* ist. Bei jedem *geraden* L kann der Fall eintreten, dass in bestimmten Komponenten der L Ausgangslösungen Nullen und Einsen gleich häufig auftreten. Im Beispiel $L = 4$ und x^1, x^2, x^3 wie oben sowie

$$x^4 = (0, 0, 1, 1)$$

erhält man in den ersten drei Komponenten klare Mehrheitsverhältnisse. In den vierten Komponenten von x^1, \dots, x^4 kommen jedoch zwei Nullen und zwei Einsen vor. Hier sollen sowohl obige Lösung $x = (0, 1, 1, 0)$ als auch $x' = (0, 1, 1, 1)$ als „Mehrheitsvektoren“ bezeichnet werden. Existieren k Komponenten mit jeweils gleich vielen Nullen und Einsen, erhält man 2^k Alternativen für x .

DEFINITION 1.1 (Mehrheitsvektor, Mehrheitsbildung)

Seien $n \geq 1$, $L \geq 2$ und $x^\ell = (x_1^\ell, x_2^\ell, \dots, x_n^\ell) \in \{0, 1\}^n$ für $\ell = 1, \dots, L$.

(i) Jeder Vektor $x = (x_1, \dots, x_n) \in \{0, 1\}^n$ mit

$$x_i = \begin{cases} 0, & \text{falls } x_i^1 + \dots + x_i^L < \frac{L}{2} \\ 1, & \text{falls } x_i^1 + \dots + x_i^L > \frac{L}{2} \end{cases}$$

für $i = 1, \dots, n$ heißt ein **Mehrheitsvektor** zu x^1, \dots, x^L .

(ii) Die zufällige Wahl eines Mehrheitsvektors gemäß der diskreten Gleichverteilung auf der Menge aller Mehrheitsvektoren zu x^1, \dots, x^L heißt **Mehrheitsbildung**. Da der Mehrheitsvektor zu einer „Gruppe“ von L Lösungen gewählt wird, soll L als **Gruppengröße** bezeichnet werden.

Ziel der vorliegenden Dissertation ist es zu untersuchen, inwiefern sich die Mehrheitsbildung in der Praxis der Kombinatorischen Optimierung bewährt, also in welchen Situationen es sinnvoll ist, mit Mehrheitsvektoren zu arbeiten.

Ist eine Maximierungsaufgabe mit $V = \{0, 1\}^n$ und $f : V \rightarrow \mathbb{R}$ gegeben, so kann man natürlich nicht erwarten, dass ein Mehrheitsvektor aus L willkürlichen Lösungen x^1, \dots, x^L einen besseren f -Wert als die Ausgangslösungen liefert. Ist etwa jede Komponente von

x^1, \dots, x^L unabhängig mit Wahrscheinlichkeit von je $\frac{1}{2}$ als 0 bzw. 1 gewählt worden, so treten 0 und 1 auch in jeder Komponente eines Mehrheitsvektors zu x^1, \dots, x^L mit gleicher Wahrscheinlichkeit auf. x^1, \dots, x^L sollten schon ein gewisses Maß an Güte mitbringen. Für die Zwecke dieser Arbeit werden lokal maximale Lösungen bezüglich der HAMMING-Nachbarschaft herangezogen:

DEFINITION 1.2 (Lokal maximale Lösung)

Seien $n \geq 1$ und $f : \{0, 1\}^n \rightarrow \mathbb{R}$. Für $x = (x_1, \dots, x_n) \in \{0, 1\}^n$ und $i \in \{1, \dots, n\}$ bezeichne $y^i(x) := (x_1, \dots, x_{i-1}, 1 - x_i, x_{i+1}, \dots, x_n)$ den *i-ten* (HAMMING-) **Nachbarn** von x . x heißt **lokal maximal**, wenn für alle $i \in \{1, \dots, n\}$ gilt:

$$f(x) \geq f(y^i(x))$$

Die Festlegung auf lokal maximale Lösungen erscheint zunächst weit her geholt. Sie birgt jedoch die Hoffnung in sich, dass die komponentenweise Optimalität mit dem komponentenweisen Mehrheitsentscheid harmoniert. Als Werkzeug zur Erzeugung einer lokal maximalen Lösung wird **lokale Suche** eingesetzt. Zur Gewinnung mehrerer lokal maximaler Lösungen wird lokale Suche entsprechend oft wiederholt. Man spricht dann von mehreren **Läufen wiederholter lokaler Suche**. Jeder Lauf beginnt mit einer bestimmten Startlösung $x \in \{0, 1\}^n$. Ein Lauf, für den x gemäß der diskreten Gleichverteilung auf $\{0, 1\}^n$ zufällig gewählt wird, soll als **zufälliger Lauf** bezeichnet werden. Startet lokale Suche dagegen in einem Mehrheitsvektor x zu lokal maximalen Lösungen x^1, \dots, x^L , soll von einem **Mehrheitslauf** gesprochen werden. Hat x einen oder mehrere HAMMING-Nachbarn mit größerem Zielfunktionswert, geht man zu einem solchen über und macht mit diesem statt mit x weiter. Welcher bessere HAMMING-Nachbar von x für diesen lokalen Verbesserungsschritt ausgewählt wird, hängt von der speziellen Variante lokaler Suche ab. Die für diese Arbeit implementierten Varianten werden in Abschnitt 1.1 beschrieben. Ist x lokal maximal, terminiert das Verfahren in x . Es ist klar, dass jeder Lauf wiederholter lokaler Suche genau eine lokal maximale Lösung liefert.

Natürlich kann man Effekte der Mehrheitsbildung auch durch Anwendung auf andere, in irgendeinem zu Definition 1.2 alternativen Sinne gute Lösungen x^1, \dots, x^L untersuchen. Eine Möglichkeit besteht darin, die Ausgangslösungen durch beliebige andere Heuristiken zu erzeugen. Denkbar ist auch, L beste Elemente aus einer größeren Menge zufällig gewählter Lösungen herauszugreifen und die Mehrheitsbildung auf diese anzuwenden. Manche Optimierungsaufgaben erlauben es sogar, nach Vorgabe einer beliebigen Güte $w \in \mathbb{R}$ auf einfache Weise ein $x \in V$ mit $f(x) = w$ zu konstruieren, falls eine solche Lösung x existiert. Dies ist beispielsweise beim künstlichen Counting-Ones-Problem möglich, bei dem jedem Vektor $x \in \{0, 1\}^n$ als Zielfunktionswert die Anzahl seiner Einsen zugeordnet wird (vgl. Abschnitt 1.2.6).

1.1 Varianten der lokalen Suche

Sei $f : \{0, 1\}^n \rightarrow \mathbb{R}$. Gesucht ist eine lokal maximale Lösung, d.h. ein Element $x \in \{0, 1\}^n$ mit $f(x) \geq f(y^i(x))$ für $i = 1, \dots, n$, wobei $y^i(x) := (x_1, \dots, x_{i-1}, 1 - x_i, x_{i+1}, \dots, x_n)$ den *i-*

ten HAMMING-Nachbarn von x bezeichnet. Zur algorithmischen Ermittlung solch einer Lösung x werden im Folgenden drei Varianten lokaler Suche vorgestellt, nämlich volle lokale Suche, schnelle lokale Suche und zyklische schnelle lokale Suche.

Für jede Variante wird ein zufälliger Lauf beschrieben, also ein Lauf mit zufällig gewähltem Startvektor. Jedes Element $x \in \{0, 1\}^n$ wird dabei mit Wahrscheinlichkeit $(2^n)^{-1}$ herangezogen. Alternativ dazu könnte die Startlösung durch spezielle, problemspezifische Heuristiken wie etwa Greedy-Strategien gewonnen werden. Um keine Seiteneffekte zu erhalten, wurden solche Verfahren für die vorliegende Arbeit nicht angewendet.

Wird im Folgenden von einer zufälligen Wahl einer Permutation $\pi \in S_n$ gesprochen, so ist die Wahl stets bezüglich der diskreten Gleichverteilung auf der Menge S_n aller Permutationen, d.h. bijektiven Abbildungen $\{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$ zu verstehen. Auf jedes $\pi \in S_n$ entfällt eine Wahrscheinlichkeit von $(n!)^{-1}$.

1.1.1 Volle lokale Suche

Die Grundidee der vollen lokalen Suche besteht darin, möglichst große lokale Verbesserungsschritte zu machen. Dazu werden erst *alle* Nachbarn der aktuell behandelten Lösung $x \in \{0, 1\}^n$ ermittelt und bewertet. Existieren Nachbarn mit besserem Zielfunktionswert, wird anschließend zu einem der *besten* Nachbarn von x übergegangen.

ALGORITHMUS 1.3 (Volle lokale Suche)

Schritt 1: Wähle $x \in \{0, 1\}^n$ zufällig.

Schritt 2: Wähle zufällig eine Permutation $\pi \in S_n$.

Schritt 3: Setze $f_{\max} := f(x)$.

Für $i = 1, \dots, n$

Falls $f(y^{\pi(i)}(x)) > f_{\max}$, setze $f_{\max} := f(y^{\pi(i)}(x))$ und $i_0 := i$.

Schritt 4: Falls $f_{\max} > f(x)$, setze $x := y^{\pi(i_0)}(x)$ und gehe zu Schritt 2.

Sonst: stopp. x ist lokal maximal.

Man beachte, dass die n HAMMING-Nachbarn des aktuellen Vektors x in Schritt 3 nicht zwingend in der kanonischen Reihenfolge $y^1(x), y^2(x), \dots, y^n(x)$, sondern in einer zufälligen Reihenfolge $y^{\pi(1)}(x), y^{\pi(2)}(x), \dots, y^{\pi(n)}(x)$ mit $\pi \in S_n$ untersucht werden. Im Fall $\{i \mid f(y^i(x)) > f(x)\} \neq \emptyset$ wird in Schritt 4 folglich zu dem gemäß der durch π festgelegten Anordnung *ersten* HAMMING-Nachbarn mit maximalem Zielfunktionswert übergegangen. Hat x insgesamt $d \in \{1, \dots, n\}$ bessere Nachbarn mit maximalem f -Wert, so wird mittels der Wahl von π zu jedem dieser Vektoren mit derselben Wahrscheinlichkeit d^{-1} gewechselt.

In der Praxis kann man oft davon ausgehen, dass die Rechenzeit eines entsprechenden Programms nahezu vollständig in die Evaluierung von Zielfunktionswerten investiert wird. Um die Effizienz verschiedener Varianten lokaler Suche zu vergleichen, kann man dann ein höchst einfaches Kostenmodell aufstellen: Man zählt die auftretenden Fragen nach der Güte von HAMMING-Nachbarn während des Programmablaufs. Diese Zahl wird als

Fragenzahl bezeichnet. Darüber hinaus sind die **Schritt**zahlen von Interesse, die die Übergänge zu benachbarten Lösungen zählen. Oft gehen kleine Schrittzahlen mit kurzen Rechenzeiten einher. Dies kann jedoch nur als äußerst grobe Faustregel verstanden werden. Die Schrittzahlen geben einen Anhaltspunkt für den HAMMING-Abstand² von Startlösung s und lokal maximaler Ziellösung t . Man beachte aber, dass lokale Suche nicht zwingend den kürzesten Weg von s nach t über sukzessive benachbarte Lösungen geht.

Ist die Startlösung eines zufälligen Laufs lokal maximal, so ist die zugehörige Schrittzahl trivialerweise Null. Trotzdem muss das Suchverfahren zum Test auf lokale Maximalität n Fragen stellen. Dies trifft auch für Mehrheitsläufe zu, sofern die lokale Maximalität nicht a priori bekannt ist. Die Ausnahme bilden Mehrheitsläufe, deren Startvektoren mit einer der lokal maximalen Lösungen x^1, x^2, \dots, x^L übereinstimmen, aus denen sie durch Mehrheitsbildung hervorgehen. Bei allen Versuchen für die vorliegende Arbeit wurden die Mehrheitsvektoren zu statistischen Zwecken auf diese Eigenschaft hin überprüft. Entsprechende Läufe lokaler Suche wurden dann in der Startlösung abgebrochen. Insbesondere wurde ihnen Schritt- und Fragenzahl 0 zugeordnet.

1.1.2 Schnelle lokale Suche

Die schnelle lokale Suche unterscheidet sich von der vollen lokalen Suche dadurch, dass sie von einer aktuellen Lösung x zu einem besseren HAMMING-Nachbarn y übergeht, sobald ein solcher entdeckt wird, *ohne* die übrigen Nachbarn von x zu testen. Die Verbesserungen sind dadurch im Durchschnitt kleiner, werden aber „spontaner“ durchgeführt.

ALGORITHMUS 1.4 (Schnelle lokale Suche)

Schritt 1: Wähle $x \in \{0, 1\}^n$ zufällig.

Schritt 2: Wähle zufällig eine Permutation $\pi \in S_n$.

Schritt 3: Für $i = 1, \dots, n$
 Falls $f(y^{\pi(i)}(x)) > f(x)$, setze $x := y^{\pi(i)}(x)$ und gehe zu Schritt 2.

Schritt 4: Stopp. x ist lokal maximal.

Der Verzicht auf die vollständige Untersuchung aller Nachbarn von x gegenüber der vollen lokalen Suche führt in der Praxis meist zu einer deutlichen Ersparnis an Rechenzeit. An einigen Stellen dieser Arbeit wird daher ganz auf den Einsatz der vollen lokalen Suche verzichtet.

Sowohl bei der vollen lokalen Suche als auch bei der schnellen lokalen Suche wird nach jedem Übergang zu einem besseren Nachbarn eine erneute Festlegung von π gefordert. Die Wahl und Speicherung einer Permutation $\pi \in S_n$ in Schritt 2 kostet immerhin linear in n viele Rechenoperationen. Andererseits hat man es nicht selten mit Optimierungsaufgaben zu tun, deren Zielfunktion f es ermöglicht, den Wert $f(y^{\pi(i)}(x))$ ausgehend von $f(x)$ mittels weniger, in ihrer Anzahl von n unabhängiger Operationen zu evaluieren. Die Kosten

²Der HAMMING-Abstand zweier Vektoren $x, y \in \{0, 1\}^n$ mit $x = (x_1, \dots, x_n)$ und $y = (y_1, \dots, y_n)$ ist definiert durch $d_H(x, y) := \#\{i \in \{1, \dots, n\} : x_i \neq y_i\}$.

für die Schritte 2 und 3 sind dann im Worst Case von der gleichen Größenordnung, was das in Abschnitt 1.1.1 beschriebene Kostenmodell in Frage stellt. Im Folgenden wird eine dritte Variante lokaler Suche beschrieben, die mit einer einzigen Wahl von π auskommt.

1.1.3 Zyklische schnelle lokale Suche

Für jeden Vektor $x \in \{0, 1\}^n$ existiert eine kanonische Reihenfolge seiner HAMMING-Nachbarn, nämlich die Reihenfolge $y^1(x), y^2(x), \dots, y^n(x)$. Man kann den i -ten Nachbarn von x gemäß dieser kanonischen Enumeration auch als Nachbarn „in Suchrichtung i “ auffassen. Die schnelle lokale Suche wird nun dahingehend modifiziert, dass die Suchrichtungen „gleichmäßig“ durchlaufen werden: Nach einer lokalen Verbesserung macht man mit einer Suchrichtung weiter, die vor dem Übergang zum Nachbarvektor möglichst lange nicht gewählt worden ist.

ALGORITHMUS 1.5 (Zyklische schnelle lokale Suche)

Sei $\pi \in S_n$ fest gewählt.

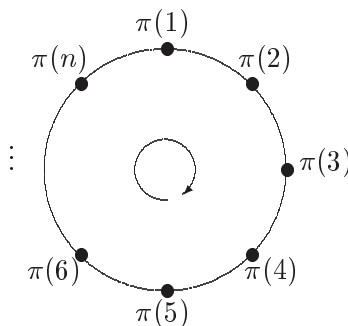
Schritt 1: Wähle $x \in \{0, 1\}^n$ zufällig.

Schritt 2: Wähle $i \in \{1, \dots, n\}$ zufällig und setze $i^* := i$.

Schritt 3: Falls $f(y^{\pi(i)}(x)) > f(x)$, setze $x := y^{\pi(i)}(x)$ und $i^* := i$.
Setze $i := (i \bmod n) + 1$.

Schritt 4: Falls $i = i^*$, stopp. x ist lokal maximal.
Sonst: Gehe zu Schritt 3.

Um die Bezeichnung „zyklisch“ zu verstehen, denke man sich die n Suchrichtungen in der durch π festgelegten Reihenfolge $\pi(1), \pi(2), \dots, \pi(n)$ auf einen Kreis angeordnet:



In den ersten beiden Schritten werden die Startlösung $x \in \{0, 1\}^n$ sowie ein Anfangsargument i^* für π festgelegt. Anschließend wird die Suche nach besseren Nachbarn von x in Suchrichtung $\pi(i^*)$ gestartet und mit $\pi(i^* + 1), \pi(i^* + 2)$ usw. fortgesetzt, wobei der Zyklus nach $\pi(n)$ mit $\pi(1)$ geschlossen wird. Falls π speziell die identische Permutation ist, werden die Nachbarn in der kanonischen Reihenfolge untersucht. Wird bei Durchlaufen des Kreises in einer Suchrichtung $\pi(i)$ ein besserer Nachbar entdeckt, geht man zu diesem über und merkt sich das entsprechende Argument i in der Variablen i^* . i^* charakterisiert also ab dem ersten Übergang zu einem Nachbarn die Stelle der jeweils letzten Verbesserung auf dem Kreis. Ist, beginnend bei i^* , eine volle Runde lang keine Verbesserung mehr

aufgetreten, nimmt i erneut den Wert i^* an, und das Verfahren terminiert in einer lokal maximalen Lösung.

1.1.4 Deterministische und probabilistische lokale Suche

Die Varianten lokaler Suche aus den Abschnitten 1.1.1 bis 1.1.3 erfordern an verschiedenen Stellen eine Zufallsentscheidung. Bei allen drei Suchvarianten wird zumindest der Startvektor zufällig gewählt. Die volle sowie die schnelle lokale Suche erzwingen zudem im Allgemeinen mehrmals eine zufällige Anordnung der Nachbarn einer aktuell behandelten Lösung. Per Definition der Algorithmen ist das Ergebnis lokaler Suche insbesondere nicht determiniert. Natürlich sind Generatoren von Pseudozufallszahlen zur Implementation der Verfahren deterministisch: Bei gleicher Initialisierung des Generators stellen sich dieselben Berechnungen ein. Die Algorithmen an sich sind jedoch probabilistischer Natur.

Dass die zyklische schnelle lokale Suche ab Schritt 3 deterministisch abläuft, liegt daran, dass die Permutation π global festgehalten wird. Natürlich lässt sich π auch bei voller oder schneller lokaler Suche nach außen ziehen. Man erhält dann — abgesehen vom Start — eine deterministische Version des entsprechenden Suchverfahrens. Man kann in diesem Sinne zwischen **probabilistischer** und **deterministischer voller** bzw. **schneller** lokaler Suche unterscheiden. Ohne diesen Zusatz sind die probabilistischen Versionen gemeint, wie sie in den Abschnitten 1.1.1 und 1.1.2 beschrieben worden sind. Es sind durchaus Situationen denkbar, in denen die deterministischen Varianten sinnvoll sind, etwa dann, wenn man gewisse Suchrichtungen bevorzugt untersuchen will und diese in π entsprechend „vornewegstellt“. So werden vorrangig Komponenten x_i einer aktuellen Lösung $x = (x_1, \dots, x_n) \in \{0, 1\}^n$ mit kleinem $\pi^{-1}(i)$ „geflippt“, d.h. durch $1 - x_i$ ersetzt. Man kann auf diese Weise Strukturen der Optimierungsaufgabe berücksichtigen, die beispielsweise auf ein Festhalten bestimmter Komponenten und Variation der übrigen Komponenten in guten Lösungen hindeuten. Das Wissen um solche Strukturen kann a priori vorliegen oder sich auf Vermutungen stützen. Es kann aber auch erst nach Ermittlung einer Liste von lokal maximalen Lösungen gewonnen werden und in lernende Verfahren mit Anpassung von π zu Laufzeiten eingehen. Im weiteren Verlauf dieser Arbeit wird lediglich die probabilistische volle bzw. schnelle lokale Suche eingesetzt.

Werden mehrere Läufe voller lokaler Suche durchgeführt, spricht man von **wiederholter voller lokaler Suche**. Entsprechend steht **wiederholte schnelle lokale Suche** bzw. **wiederholte zyklische schnelle lokale Suche** für die iterierte Version des entsprechenden Typs lokaler Suche. An einigen Stellen wird zwischen **deterministisch** und **probabilistisch wiederholter** lokaler Suche unterschieden. Diese Unterscheidung bezieht sich nicht etwa wie oben auf einen bestimmten Lauf lokaler Suche, sondern auf die Art der Wiederholung. Das ist bei allen deterministischen Varianten lokaler Suche sinnvoll, d.h. bei allen Verfahren, die die Permutation π innerhalb eines Laufs fest beibehalten. Im deterministischen Fall der Wiederholung bleibt π für alle Läufe fest, im probabilistischen Fall wird π vor jedem Lauf erneut zufällig bestimmt. Da in dieser Arbeit deterministische volle bzw. schnelle lokale Suche keinen Einsatz findet, tritt diese Begriffsbildung nur bei wiederholter zyklischer schneller lokaler Suche auf.

1.1.5 Lokale Suche mit Sidesteps

Bei der Masse der praktisch relevanten kombinatorischen Optimierungsaufgaben sind unter gängigen Nachbarschaftsmodellen benachbarte Lösungen mit gleichem Zielfunktionswert keine Seltenheit. Oft treten sogar ganze Plateaus gleicher f -Werte auf. Man kann dann die Güte der durch lokale Suche gefundenen Lösungen in der Regel drastisch erhöhen, wenn man auch Übergänge zu gleichwertigen Nachbarn erlaubt. Solche Übergänge sind anschaulich Seitwärtsschritte im Funktionsgebirge über dem Lösungsraum $\{0, 1\}^n$ und werden daher als **Sidesteps** bezeichnet. Bei naiven Implementationen dieser Grundidee hat man allerdings damit zu rechnen, dass Zyklen auftreten in dem Sinne, dass innerhalb eines Laufs lokaler Suche eine bestimmte Lösung mehrfach besucht wird. Zur Gewährleistung, dass das Verfahren terminiert, sind verschiedene Methoden möglich:

1. Die Schrankenmethode:

Man kann die Zahl erlaubter Sidesteps durch eine Konstante beschränken, und zwar einerseits die während eines Laufs erlaubte Gesamtzahl und/oder andererseits ihre Anzahl zwischen zwei Verbesserungsschritten. Diese Methode ist sehr einfach zu implementieren. Sie setzt aber eine passend gewählte Schranke voraus und bewirkt möglicherweise, dass die lokale Suche — im Extremfall — zwischen zwei Lösungen x und y so lange hin- und herpendelt, bis die Schranke erreicht ist. Statt die Schranke konstant zu halten, kann man sie auch variabel an Größen wie die bisher verbrauchte Rechenzeit oder den aktuellen Zielfunktionswert anpassen.

2. Die Bitfeldmethode:

Ist n nicht zu groß, bietet sich eine elegante Alternative zur Schrankenmethode an. Man merkt sich in einem Bitfeld der Länge n , in welchen Suchrichtungen seit der zuletzt aufgetretenen Verbesserung Sidesteps durchgeführt worden sind. Erlaubt werden dann nur solche Suchrichtungen für weitere Sidesteps, deren entsprechendes Bit noch nicht gesetzt ist. Beim nächsten Übergang zu einem besseren Nachbarn werden alle Bits wieder gelöscht. Man schränkt auf diese Weise die Möglichkeiten für Seitwärtsbewegungen im Vergleich zur Schrankenmethode zwar enorm ein, verhindert dafür aber ein Zirkulieren des Suchverfahrens. Zudem werden Plateaus bezüglich der Suchrichtungen gleichmäßiger und damit im Allgemeinen auch gründlicher durchlaufen. Oft hat man bei lokaler Suche, die sich lange auf einem Plateau bewegt, die Anschauung, dass das Plateau sehr „weit“ ist und nur wenige „Aufgänge“ zu besseren Lösungen existieren. Gerade um solche „Treppchen“ aufzuspüren, muss im Allgemeinen ein möglichst großer Teil des Plateaus abgesprochen werden. Falls n sehr groß ist, wird die Verwaltung des Bitfelds entsprechend aufwendig. Dann kann man auf die Schrankenmethode zurückgreifen, die neben der Konstante nur Speicherplatz für einen Zähler der Sidesteps benötigt und deren technischer Aufwand sowohl bei der Programmierung als auch zu Laufzeiten zu vernachlässigen ist.

3. Die Rundenmethode:

Bei der zyklischen schnellen lokalen Suche kann man folgende Vereinbarung treffen: Zwischen zwei Verbesserungen seien nur während einer festen Zahl k von Runden der Suche Sidesteps zugelassen. Dies erfordert lediglich einen Rundenzähler, der unproblematisch zu realisieren ist. Eine Runde beginnt und endet jeweils mit Suchrichtung

$\pi(i^*)$ (vgl. Algorithmus 1.5). Während einer Runde werden die n Suchrichtungen in der Reihenfolge $\pi(i^* + 1), \pi(i^* + 2), \dots, \pi(n), \pi(1), \pi(2), \dots, \pi(i^*)$ durchlaufen. Tritt zwischendurch, etwa in Suchrichtung $\pi(i)$, eine echte f -Wert-Verbesserung ein, wird i^* auf i und der Rundenzähler auf 1 gesetzt. Ansonsten wird die Runde mit der Suchrichtung $\pi(i^*)$ der letzten vorangegangenen Verbesserung abgeschlossen, und eine neue Runde beginnt. In diesem Fall wird der Rundenzähler um 1 erhöht. Erreicht der Zähler den Wert $k + 1$, wird die Suche abgebrochen.

Bei der Schranken- bzw. Bitfeldmethode in Verbindung mit schneller bzw. zyklischer schneller lokaler Suche kann man ferner folgende Strategien unterscheiden:

- (a) Sidesteps sind erst nach Erreichen einer lokal maximalen Lösung zugelassen. Man sucht also erst nach Festlaufen in einer solchen Lösung nach neuen „Aufgängen“, ausgehend von gleichwertigen Nachbarn. Nach einem durchgeführten Sidestep werden Seitwärtsschritte bis auf Weiteres wieder verboten. Der Einsatz von Sidesteps dient also als Versuch, aus lokal maximalen Lösungen ohne f -Wert-Verschlechterung heraus zu gelangen. Bei der vollen lokalen Suche tritt diese Vorgehensweise automatisch in Kraft.
- (b) Sidesteps sind prinzipiell an beliebiger Stelle zugelassen, so lange das Abbruchkriterium noch nicht erfüllt ist. Sind, ausgehend von einer Lösung x , Sidesteps erlaubt und tritt irgendwann später eine erste f -Wert-Verbesserung zu einem Vektor y auf, so liegen auf dem Pfad von x nach y im Allgemeinen Lösungen, die selbst nicht lokal maximal sind. In diesem Sinne wird ein Vorbeilaufen an „Aufgängen“ zu besseren Lösungen nicht vermieden. Man bewegt sich aber auf einem Plateau schneller als in (a) voran, da nicht nach jedem Seitwärtsschritt der komplette Aufwand für die Untersuchung aller Nachbarn anfällt. Diese Strategie wird bei der Rundenmethode automatisch angewendet.
- (c) Man kann auch eine Mischform aus (a) und (b) zulassen: Zunächst wird ohne Sidesteps eine lokal maximale Lösung x ermittelt. Ab Erreichen von x sind dann so lange Sidesteps erlaubt, bis eine echte Verbesserung zu einer Lösung y eintritt. Ausgehend von y wird eine neue lokal maximale Lösung gesucht, ohne Sidesteps zuzulassen. Der Vorgang wird so lange wiederholt, bis das Abbruchkriterium erfüllt ist.

Man beachte, dass in all diesen Varianten lokaler Suche mit Sidesteps nicht gewährleistet ist, dass das Verfahren in einer lokal maximalen Lösung terminiert. Es kann vorkommen, dass gegen Ende des Ablaufs nur noch Seitwärtsschritte gemacht werden, bis das jeweilige Abbruchkriterium aus 1. bis 3. erfüllt ist. Der Abbruch kann erfolgen, bevor alle Nachbarn der letzten Lösung getestet worden sind. In diesem Fall sollte eine Überprüfung auf lokale Maximalität angefügt werden.

In den Versuchen für die vorliegende Dissertation wurden alle Varianten (1.-3., (a)-(c)) getestet. Bei den betrachteten Problemen stellte sich heraus, dass die generelle Erlaubnis von Sidesteps in Läufen lokaler Suche zu deutlich besseren Lösungen führt (vgl. Abschnitte 2.1.1, 4.1, 4.2.1 und A 2.3). Welche spezielle Variante eingesetzt wird, spielt dagegen eine untergeordnete Rolle. Die Schranke bei der Schrankenmethode bzw. die Rundenzahl k bei der Rundenmethode sollten aber großzügig gewählt werden. Die Beschreibungen zu

Versuchen mit Sidesteps beziehen sich stets auf Variante 1. mit (b), welche besonders einfach zu programmieren ist.

1.2 Untersuchte Optimierungsprobleme

Im Folgenden werden die Probleme mit Lösungsraum $\{0, 1\}^n$ vorgestellt, bei denen die Mehrheitsbildung untersucht werden soll. Die Instanzen einiger dieser Probleme sind **Minimierungsaufgaben** statt Maximierungsaufgaben, d.h. man sucht nach einer Lösung $x \in \{0, 1\}^n$ mit möglichst kleinem statt möglichst großem Zielfunktionswert. Um die verschiedenen Varianten lokaler Suche aus Abschnitt 1.1 anwenden zu können, ersetze man in solchen Fällen die Zielfunktion f durch $-f$. Eine Lösung $x \in \{0, 1\}^n$ für eine Minimierungsaufgabe mit Zielfunktion f heißt **lokal minimal**, wenn x bezüglich der Maximierungsaufgabe mit Zielfunktion $-f$ nach Definition 1.2 lokal maximal ist.

1.2.1 Max-SAT

Der „Urvater“ der NP-vollständigen Probleme ist das **Satisfiability-Problem**, kurz **SAT-Problem**. Hier sind als Instanz n BOOLEsche Variable X_1, \dots, X_n und eine BOOLEsche Formel in konjunktiver Normalform (KNF) vorgegeben. Es ist zu entscheiden, ob eine Belegung $x = (x_1, \dots, x_n) \in \{0, 1\}^n$ der Variablen mit möglichen Werten 1 („wahr“) oder 0 („falsch“) existiert, die die Formel erfüllt. Die einfachsten BOOLEschen Formeln sind X_i bzw. $\neg X_i$, also eine Variable oder ihre Negation. Eine Variable X_i ist genau dann erfüllt, wenn sie mit 1 belegt ist, d.h. wenn $x_i = 1$ gilt. Ihre Negation $\neg X_i$ ist genau dann erfüllt, wenn X_i nicht erfüllt, also $x_i = 0$ ist. Diese beiden Arten einer BOOLEschen Formel bezeichnet man als **Literale**. Verknüpft man endlich viele Literale L_1, \dots, L_k disjunktiv, so spricht man von einer **Klausel**. Als Verknüpfungszeichen zwischen je zwei Literalen dient das Symbol „ \vee “. Eine Klausel $(L_1 \vee \dots \vee L_k)$ ist genau dann erfüllt, wenn *mindestens* ein Literal L_j erfüllt ist. Eine BOOLEsche Formel ist in **KNF**, wenn sie aus m konjunktiv, d.h. symbolisch mittels „ \wedge “ verknüpften Klauseln besteht. Sie ist genau dann erfüllt, wenn *jede* Klausel erfüllt ist.

BEISPIEL. Seien $n = 3$ und $m = 4$. Dann ist zur Erfüllung der BOOLEschen Formel

$$F = (X_1 \vee X_3) \wedge (\neg X_2 \vee X_3 \vee X_1) \wedge (\neg X_1) \wedge (X_2 \vee X_1 \vee \neg X_3)$$

die Bedingung $x_1 = 0$ notwendig, da die dritte Klausel nur dann erfüllt ist. Also kann die erste Klausel nur erfüllt sein, wenn zudem $x_3 = 1$ ist. Die zweite Klausel ist nun wegen $x_3 = 1$ unabhängig von der Belegung von X_2 erfüllt. Erst die Erfüllung der vierten Klausel erfordert $x_2 = 1$. Die Belegung $x = (0, 1, 1)$ erfüllt somit als einzige die Formel F .

Man kann leicht SAT-Instanzen konstruieren, die mehrere erfüllende Belegungen besitzen, oder auch solche, für die es keine erfüllende Belegung gibt. Letztlich entscheidend ist nur, ob die Formel generell erfüllbar ist oder nicht. COOK [10] hat 1971 gezeigt, dass das SAT-Problem NP-vollständig ist. Heute geht man deshalb davon aus, dass für kein Polynom p

ein deterministischer Algorithmus existiert, der nach Eingabe einer beliebigen, kompakt mittels N Bits kodierten Instanz mit BOOLEscher Formel F in KNF in $p(N)$ Schritten entscheidet, ob F erfüllbar ist oder nicht.

Eine spezielle Teilklasse der SAT-Instanzen bilden diejenigen mit Formeln in KNF, deren Klauseln allesamt dieselbe **Länge**, d.h. dieselbe Zahl an Literalen haben. Beim Problem k -SAT hat jede Klausel genau k Literale. Sowohl eines Literals als auch einer Variable innerhalb einer Klausel sind Wiederholungen erlaubt. In [10] wird gezeigt, wie sich jede SAT-Instanz mit polynomielltem Zeitaufwand in eine 3-SAT-Instanz transformieren lässt. Durch Einfügen von Kopien eines Literals in eine Klausel kann man zudem 3-SAT-Instanzen auf k -SAT-Instanzen mit $k \geq 3$ bringen. Da k -SAT der Klasse NP angehört, ist es folglich wie schon SAT NP-vollständig, und zwar für jedes $k \geq 3$.

Die Optimierungs-Variante von k -SAT wird als **Max- k -SAT** bezeichnet. Gegeben sind als Instanz n BOOLEsche Variable X_1, \dots, X_n und eine Liste von m Klauseln, je der Länge k , über diesen Variablen. Ziel ist es, eine Belegung $x \in \{0, 1\}^n$ zu finden, die möglichst viele Klauseln erfüllt. Die Zahl der unter x erfüllten Klauseln sei $f(x)$. Dann ist dadurch eine Maximierungsaufgabe mit Lösungsraum $\{0, 1\}^n$ und Zielfunktion f definiert.

BEISPIEL. Seien $k = 2$, $n = 3$, $m = 4$ und folgende Klauseln gegeben:

$$(X_1 \vee X_2) \quad (\neg X_1 \vee X_2) \quad (\neg X_3 \vee \neg X_3) \quad (\neg X_2 \vee X_3)$$

Man sieht sofort, dass $f(x) = 3$ für $x = (1, 1, 0)$ gilt. Ermittelt man die Zielfunktionswerte aller acht möglichen Lösungen, so stellt man fest, dass der f -Wert 3 maximal ist.

GAREY, JOHNSON und STOCKMEYER [17] haben 1976 bewiesen, dass Max-2-SAT NP-vollständig ist, indem sie eine Polynomzeit-Reduktion des Problems 3-SAT auf das zu Max-2-SAT gehörende Entscheidungsproblem angegeben haben. Bei diesem soll unter zusätzlicher Eingabe einer Schranke s entschieden werden, ob es eine Belegung gibt, die mindestens s Klauseln erfüllt. Mit Max-2-SAT ist Max- k -SAT für beliebiges $k \geq 2$ NP-vollständig. Dies wird wiederum durch Einfügen von Kopien eines Literals innerhalb einer Klausel offensichtlich. Ist die Länge der Klauseln variabel, spricht man vom Problem **Max-SAT**. Als Verallgemeinerung von Max- k -SAT ist auch Max-SAT NP-vollständig.

1.2.2 Ein Dynamisches Optimierungsproblem

Die NP-Vollständigkeit der im vorangegangenen Abschnitt vorgestellten Probleme Max-SAT bzw. Max- k -SAT rechtfertigt den Einsatz von Heuristiken zur Ermittlung suboptimaler Lösungen. Das soll aber nicht heißen, dass lokale Suche und Mehrheitsbildung nur auf solche Probleme anwendbar sind. Eine Untersuchung von Mehrheitsbildung trägt auch bei vergleichsweise einfachen Optimierungsaufgaben zur besseren Einschätzung ihrer Wirkungsweise bei.

Eine Klasse von Problemen mit gut handhabbaren, exakt lösbaren Aufgaben ist die Klasse der **Dynamischen Optimierungsprobleme**. In der einfachsten Version sind hier als

Instanz reellwertige Matrizen

$$C(t) = (c_{ij}(t))_{\substack{i=0,1 \\ j=0,1}} = \begin{pmatrix} c_{00}(t) & c_{01}(t) \\ c_{10}(t) & c_{11}(t) \end{pmatrix}$$

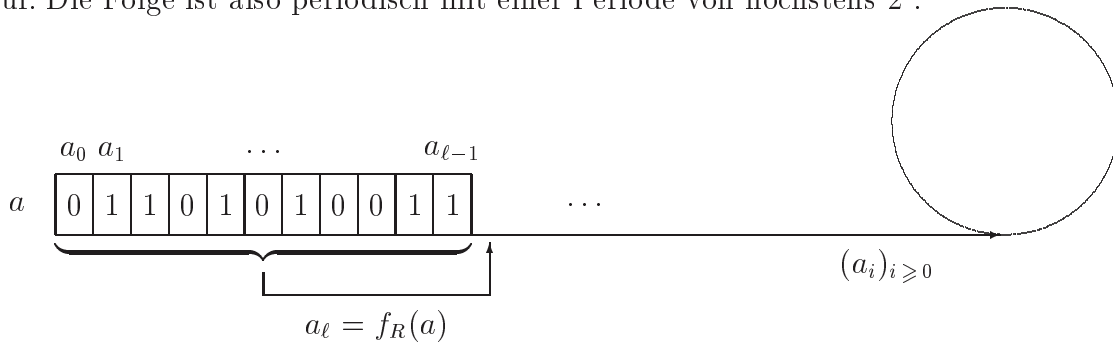
für $t = 1, 2, \dots, n-1$ gegeben. Gesucht ist ein Vektor $x = (x_1, \dots, x_n) \in \{0, 1\}^n$, für den

$$f(x) := \sum_{t=1}^{n-1} c_{x_t x_{t+1}}(t)$$

größtmöglich ist. Für diese Maximierungsaufgabe sind exakte Verfahren bekannt, die in $O(n)$ Schritten durch **Vorwärts-** bzw. **Rückwärtsrekursion** sukzessive global maximale Lösungen $x \in \{0, 1\}^n$ konstruieren. Diese Idee der **Dynamischen Optimierung** geht auf BELLMAN zurück [7].

1.2.3 Ein Shiftregister-Problem

Man betrachte ein **Shiftregister** der Länge ℓ , also ein Register, in dem ℓ Bits gespeichert werden und eine aktuelle Belegung $a = (a_0, \dots, a_{\ell-1}) \in \{0, 1\}^\ell$ durch eine der Nachfolge-Belegungen $a^{<b} := (a_1, \dots, a_{\ell-1}, b)$ mit $b \in \{0, 1\}$ ersetzt wird. $a^{<0}$ bzw. $a^{<1}$ geht also aus a durch Links-Shiften um eine Position unter Wegfall von a_0 und Anhängen des Bits 0 bzw. 1 hervor. Welches Bit angehängt wird, ist durch die sogenannte **Rückkopplungsfunktion** $f_R : \{0, 1\}^\ell \rightarrow \{0, 1\}$ bestimmt: es gelte $b = f_R(a)$. Startet man etwa mit einer festen Anfangsbelegung $(a_0, \dots, a_{\ell-1}) \in \{0, 1\}^\ell$, ersetzt diese durch (a_1, \dots, a_ℓ) mit $a_\ell = f_R(a_0, \dots, a_{\ell-1})$, im nächsten Schritt durch $(a_2, \dots, a_{\ell+1})$ mit $a_{\ell+1} = f_R(a_1, \dots, a_\ell)$, so erhält man durch weiteres Iterieren schließlich eine Bitfolge $(a_i)_{i \geq 0}$ mit $a_i = f_R(a_{i-\ell}, \dots, a_{i-1})$ für $i \geq \ell$. $(a_i)_i$ heißt **Shiftregisterfolge**. Da nur 2^ℓ verschiedene Registerbelegungen möglich sind, tritt nach spätestens 2^ℓ Iterationen eine Wiederholung der aktuellen Belegung auf. Die Folge ist also periodisch mit einer Periode von höchstens 2^ℓ .



Shiftregister finden in der Kodierungstheorie, beim Entwurf von Pseudo-Zufallszahlen-Generatoren oder in der Kryptographie Anwendung. Interessant sind etwa Rückkopplungsfunktionen, bei denen unter jeder Anfangsbelegung $a = (a_0, \dots, a_{\ell-1}) \in \{0, 1\}^\ell$ die Periode $p(a)$ der resultierende Folge $(a_i)_i$ möglichst groß ist. Zudem sollen die Folgen $(a_i)_i$ gewisse statistische Eigenschaften erfüllen. Dazu gehören beispielsweise annähernd gleiche Häufigkeiten von Nullen und Einsen innerhalb des Periodenstücks.³

³Eine ausführliche theoretische Behandlung von Shiftregistern findet man in [21].

Im Folgenden sei die Rückkopplungsfunktion f_R speziell definiert durch

$$f_R(x_0, \dots, x_{\ell-1}) := AM(x_{\ell-v_1}, \dots, x_{\ell-v_k}),$$

wobei $\{v_1, \dots, v_k\}$ eine k -elementige Teilmenge von $\{1, \dots, \ell\}$ für ein festes ungerades $k \leq \ell$ sei und

$$AM : \{0, 1\}^k \rightarrow \{0, 1\}, \quad AM(y_1, \dots, y_k) := \begin{cases} 0, & \text{falls } \sum_{i=1}^k y_i > \frac{k}{2}, \\ 1, & \text{falls } \sum_{i=1}^k y_i < \frac{k}{2}, \end{cases}$$

die **Anti-Mehrheit** aus k Bits bildet. $f_R(x_0, \dots, x_{\ell-1})$ ist also dasjenige Bit, welches unter den k durch $\ell - v_1, \dots, \ell - v_k$ indizierten Komponenten von $(x_0, \dots, x_{\ell-1})$ in der Minderheit vorkommt. Da k als ungerade vorausgesetzt wird, ist die Anti-Mehrheit eindeutig festgelegt. Eine mögliche Aufgabenstellung besteht darin, die Periode $p(a)$, $a \in \{0, 1\}^\ell$, nach Vorgabe von ℓ , k und v_1, \dots, v_k zu maximieren.

Da die Evaluierung der Werte $p(a)$ mit hohem Aufwand verbunden ist, soll für die Zwecke der vorliegenden Arbeit eine einfachere, wenn auch weniger praxisrelevante Fragestellung untersucht werden. Gegeben seien die Shiftregisterlänge ℓ , die Rückkopplungsfunktion f_R mittels k und v_1, \dots, v_k wie oben sowie eine natürliche Zahl $n > k$. Interessant ist, ob n als (Vielfaches der) Periode einer entsprechenden Shiftregisterfolge $(a_i)_i$ auftritt. Dies ist genau dann der Fall, wenn es ein Anfangsstück $a = (a_0, \dots, a_{\ell-1}) \in \{0, 1\}^\ell$ gibt, so dass für die zugehörige Shiftregisterfolge $(a_i)_{i \geq 0}$ ein Index $i_0 \geq 0$ existiert mit $a_{i_0+n} = a_{i_0}$, d.h. eine endliche Teilfolge

$$(x_0, x_1, \dots, x_{n-1}) := (a_{i_0}, a_{i_0+1}, \dots, a_{i_0+n-1}) \in \{0, 1\}^n$$

mit

$$x_i = AM(x_{(i-v_1) \bmod n}, \dots, x_{(i-v_k) \bmod n})$$

für $i = 0, \dots, n-1$. Definiert man $f : \{0, 1\}^n \rightarrow \mathbb{R}$ durch

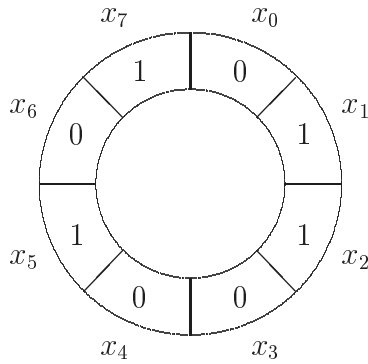
$$f(x_0, \dots, x_{n-1}) := \#\{i \mid x_i \neq AM(x_{(i-v_1) \bmod n}, \dots, x_{(i-v_k) \bmod n})\},$$

so zählt f die Komponenten jedes Vektors $x = (x_0, \dots, x_{n-1})$, die dieser Eigenschaft widersprechen. $f(x)$ heißt der **Defekt** von x . Die Vektorlänge n ist genau dann Vielfaches der Periode einer Shiftregisterfolge $(a_i)_{i \geq 0}$, wenn $\min f = 0$ gilt.

Damit ist die zu behandelnde Optimierungsaufgabe mit Lösungsraum $V = \{0, 1\}^n$ und Zielfunktion $f : V \rightarrow \mathbb{R}$ festgelegt. Um sich der Einfachheit halber von der Shiftregisterlänge ℓ zu lösen, seien $v_1, \dots, v_k \in \{1, \dots, n-1\}$ erlaubt. Eine Probleminstanz besteht dann aus natürlichen Zahlen n und k , wobei k ungerade sei und $1 \leq k < n$ gelte, sowie paarweise verschiedenen $v_1, \dots, v_k \in \{1, \dots, n-1\}$.

BEISPIELE.

a) Seien $n = 8$, $k = 3$, $v = (v_1, v_2, v_3) = (1, 2, 5)$ und $x = (0, 1, 1, 0, 0, 1, 0, 1)$. Dann gilt:



$$\begin{aligned}
 x_0 &= 0 \neq AM(1, 0, 0) = AM(x_7, x_6, x_3) \\
 x_1 &= 1 = AM(0, 1, 0) = AM(x_0, x_7, x_4) \\
 x_2 &= 1 \neq AM(1, 0, 1) = AM(x_1, x_0, x_5) \\
 x_3 &= 0 = AM(1, 1, 0) = AM(x_2, x_1, x_6) \\
 x_4 &= 0 = AM(0, 1, 1) = AM(x_3, x_2, x_7) \\
 x_5 &= 1 = AM(0, 0, 0) = AM(x_4, x_3, x_0) \\
 x_6 &= 0 = AM(1, 0, 1) = AM(x_5, x_4, x_1) \\
 x_7 &= 1 \neq AM(0, 1, 1) = AM(x_6, x_5, x_2)
 \end{aligned}$$

Also hat x Defekt $f(x) = 3$. Man überzeugt sich leicht, dass $x' = (0, 1, 0, 1, 0, 1, 0, 1)$ dagegen Defekt 0 besitzt. Es ist also $\min f = 0$.

- b) Seien $n = 6$, $k = 3$ und $v = (v_1, v_2, v_3) = (1, 2, 4)$. Dann gilt $\min f = 1$. Man kann dies etwa durch Enumeration der $2^6 = 64$ möglichen Lösungen und f -Wert-Berechnung zeigen. Beispielsweise ist $f(0, 0, 1, 1, 0, 1) = 1$.

1.2.4 Ein Ising-Spincas-Problem

Ein klassisches Problem in der Physik ist das folgende: Gegeben sei ein System von N magnetischen Momenten, sogenannten **Spins**, in fester Anordnung, etwa als Punkte eines ein- oder mehrdimensionalen Gitters. Ein Spin S_i kann in Abhängigkeit von Stärke und Drehsinn mit einer reellen Zahl entsprechenden Betrags und Vorzeichens identifiziert werden. Jeder Spin interagiert mit seinen nächsten Nachbarn. Man modelliert eine Interaktion zwischen Spin S_i und S_j durch eine Zahl $J_{ij} \in \mathbb{R}$ und setzt $J_{ij} := 0$, falls S_i und S_j nicht nächste Nachbarn sind. S_i und S_j **erfüllen** die Interaktion J_{ij} , falls gilt:

$$J_{ij} S_i S_j \geq 0$$

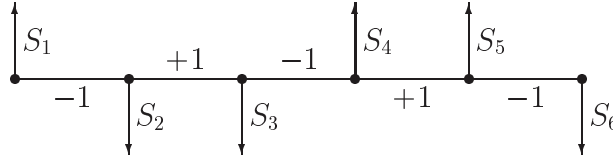
Besonders hervorzuheben sind folgende Spezialfälle: Ein **Ising-Spincas** liegt vor, wenn $S_i \in \{-1, +1\}$ für $i = 1, \dots, N$ gilt, die Spins also sämtlich normiert sind und sich höchstens in ihrem Drehsinn unterscheiden. Falls die Interaktionen J_{ij} zwischen nächsten Nachbarn S_i und S_j betragsmäßig gleich und von Null verschieden sind, etwa $|J_{ij}| = J$ mit $J > 0$, spricht man vom $\pm J$ -Modell. Liegen beide Bedingungen vor, hat man es entsprechend mit einem $\pm J$ -Ising-Modell zu tun. In Ising-Modellen ist beispielsweise folgende Optimierungsaufgabe sinnvoll: Gegeben seien Interaktionen $J_{ij} \in \mathbb{R}$ für alle $i, j \in \{1, \dots, N\}$ mit $i \neq j$. Finde $S_1, \dots, S_N \in \{-1, +1\}$, so dass

$$f(S_1, \dots, S_N) := - \sum_{i \neq j} J_{ij} S_i S_j$$

minimal ist. Die Lösungen $(S_1, \dots, S_N) \in \{-1, +1\}^N$ heißen **Spinzustände**. $f(S_1, \dots, S_N)$ wird als **Energie** des Spinzustands (S_1, \dots, S_N) bezeichnet. Ein Spinzustand mit kleinstmöglicher Energie heißt **Grundzustand**. Gelegentlich erweitert man das Modell durch Einbeziehen eines äußeren Magnetfelds der Stärke $h > 0$ und definiert dann

$$f(S_1, \dots, S_N) := - \sum_{i \neq j} J_{ij} S_i S_j - h \sum_i S_i.$$

BEISPIEL. Betrachte das ± 1 -Ising-Modell auf einem Pfad mit $N = 6$ ohne äußeres Feld. Es gelte $J_{12} = J_{34} = J_{56} = -1$ sowie $J_{23} = J_{45} = +1$ und $J_{ij} = 0$ sonst.



Offensichtlich erfüllt $S = (+1, -1, -1, +1, +1, -1)$ alle Interaktionen. Es gilt $f(S) = -5$. Ferner gibt es genau zwei Grundzustände, nämlich S und $-S = (-1, +1, +1, -1, -1, +1)$. Kommt ein äußeres Feld der Stärke $h = 2$ dazu, gilt

$$f(+1, +1, +1, +1, +1, +1) = -2 + 3 - 2 \cdot 6 = -11.$$

Man überzeugt sich leicht, dass -11 die kleinstmögliche Energie ist.

Wie das ± 1 -Ising-Modell lässt sich offensichtlich jedes $\pm J$ -Ising-Modell auf Pfaden ohne äußeres Magnetfeld „brute force“ optimieren. Es handelt sich um einen Spezialfall des in Abschnitt 1.2.2 vorgestellten Dynamischen Optimierungsproblems. Setze dazu

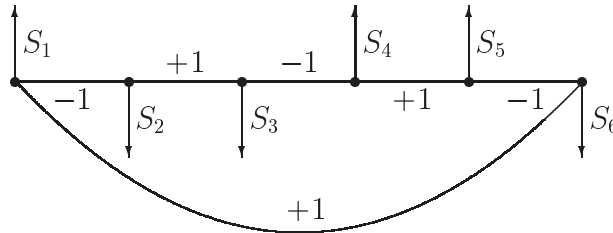
$$c_{ij}(t) := J_{t,t+1} (2i - 1)(2j - 1), \quad i, j = 0, 1, \quad t = 1, 2, \dots, N - 1.$$

Dann ist (S_1, \dots, S_N) genau dann Grundzustand, wenn (x_1, \dots, x_N) mit $x_t := \frac{1}{2}(S_t + 1)$ für $t = 1, \dots, N - 1$ eine maximale Lösung für diese Instanz des Dynamischen Optimierungsproblems ist. Es gilt nämlich

$$f(x_1, \dots, x_N) = - \sum_{t=1}^{N-1} J_{t,t+1} S_t S_{t+1} = - \sum_{t=1}^{N-1} c_{x_t x_{t+1}}(t).$$

Im Zusammenhang mit Ising-Spिंगläsern wird daher auf die Behandlung der eindimensionalen Modelle verzichtet.

Wesentlich komplizierter zu optimieren sind Spinglas-Modelle, deren Interaktionen nicht sämtlich erfüllbar sind. Interagiert etwa im obigen Pfad-Beispiel zusätzlich Spin S_6 mit S_1 und gilt $J_{61} = +1$, so wird J_{61} von S nicht erfüllt:



Es gilt $f(S) = -5 + 1 = -4$. Trotz der nicht erfüllten Interaktion bleibt S Grundzustand, da offensichtlich nicht jede Interaktion erfüllbar ist. Da man Kreise insbesondere in einen Torus einbetten kann, existieren auch für den Torus Problem-Instanzen, deren Interaktionen nicht allesamt erfüllt werden können. BARAHONA [5] hat 1981 für den $(n \times n)$ -Torus gezeigt, dass sich im Falle des $\pm J$ -Ising-Modells ohne äußeres Feld Grundzustände in $O(n^{14})$ Schritten berechnen lassen. Dieses Problem liegt also in P, der Algorithmus ist aber

nicht effizient. Kommt zusätzlich ein äußeres Feld hinzu, so ist das Problem bereits NP-schwer. Im Ising-Modell ohne äußeres Feld und unter zufällig gemäß der GAUSS-Verteilung gewählten Interaktionen auf dem Torus ist bis heute offen, ob sich alle Probleminstanzen in höchstens polynomiell vielen Schritten lösen lassen. Mit äußerem Feld ist auch dieses Problem NP-schwer (BARAHONA [6] 1993).⁴

Im Folgenden wird ausschließlich der $(n \times n)$ -Torus mit $n \geq 3$ und dem ± 1 -Ising-Modell ohne äußeres Feld behandelt. Jeder Gitterpunkt $v = (i, j) \in T := \{0, \dots, n-1\}^2$ des Torus ist mit einem Spin S_v besetzt, der mit den vier Spins S_{v_1} , S_{v_2} , S_{v_3} und S_{v_4} auf seinen nächsten Nachbarpositionen

$$\begin{aligned} v_1 &= (i, (j-1) \bmod n), & v_2 &= (i, (j+1) \bmod n), \\ v_3 &= ((i-1) \bmod n, j), & v_4 &= ((i+1) \bmod n, j) \end{aligned}$$

interagiert. Ein **Spinzustand** ist hier eine $(n \times n)$ -Matrix $(S_v)_{v \in T}$ mit Komponenten in $\{-1, +1\}$, also Element von $V := \{-1, +1\}^T$. Zwei Spinzustände $(S_v)_v, (S'_v)_v \in V$ heißen **HAMMING-benachbart**, wenn sie sich in genau einer Position $v \in T$ unterscheiden. Vermöge der bijektiven Abbildung $\varphi : V \rightarrow \{0, 1\}^N$ mit $N = n^2$, definiert durch

$$\begin{aligned} \varphi((S_v)_{v \in T}) := & \left(\begin{array}{cccc} \frac{1}{2} S_{(0,0)} + 1, & \frac{1}{2} S_{(0,1)} + 1, & \dots, & \frac{1}{2} S_{(0,n-1)} + 1, \\ \frac{1}{2} S_{(1,0)} + 1, & \frac{1}{2} S_{(1,1)} + 1, & \dots, & \frac{1}{2} S_{(1,n-1)} + 1, \\ & \vdots & & \\ \frac{1}{2} S_{(n-1,0)} + 1, & \frac{1}{2} S_{(n-1,1)} + 1, & \dots, & \frac{1}{2} S_{(n-1,n-1)} + 1 \end{array} \right), \end{aligned}$$

lässt sich die Spinglas-Aufgabe auf dem Torus in eine Minimierungsaufgabe mit Lösungsraum $\{0, 1\}^N$ überführen und mit lokaler Suche in $\{0, 1\}^N$ bearbeiten. Eine Lösung $x \in \{0, 1\}^N$ ist bzgl. der HAMMING-Nachbarschaft über $\{0, 1\}^N$ genau dann lokal minimal, wenn $\varphi^{-1}(x) \in V$ bzgl. der HAMMING-Nachbarschaft über V lokal minimal ist.

1.2.5 Ein Rucksack-Problem

Gegeben seien zwei Vektoren $s = (s_1, \dots, s_n)$ und $v = (v_1, \dots, v_n)$ mit Komponenten in $\mathbb{N}^+ := \{1, 2, \dots\}$ sowie eine Zahl $S \in \mathbb{N}^+$. Gesucht ist ein Vektor $x = (x_1, \dots, x_n) \in \{0, 1\}^n$ mit

$$s(x) := \sum_{i=1}^n x_i s_i \leq S, \tag{*}$$

für den

$$v(x) := \sum_{i=1}^n x_i v_i$$

möglichst groß ist. Man kann anschaulich von n **Paketen** sprechen und s_i bzw. v_i für $i = 1, \dots, n$ als **Größe** bzw. **Wert** des i -ten Pakets interpretieren. Die Zahl S stehe für die Größe eines **Rucksacks**, der mit einem Teil der Pakete bepackt werden soll. Jede

⁴Eine detaillierte Problembeschreibung sowie Branch-and-Cut-Ansätze für den $(n \times n)$ -Torus mit kleinen Parametern n findet man beispielsweise in [12], [13] und [24].

mögliche Rucksackladung wird dann durch einen Vektor $x \in \{0, 1\}^n$ repräsentiert, wobei $x_i = 1$ genau dann gelte, wenn das i -te Paket in den Rucksack genommen wird. Mit dieser Interpretation der Ausgangsdaten besagt Bedingung (\star) , dass die gemäß x ausgewählten Pakete zusammen in den Rucksack passen müssen. $v(x)$ gibt den Gesamtwert der Rucksackladung x wieder. Der Lösungsraum dieser Maximierungsaufgabe ist also die Menge X aller (\star) erfüllenden Vektoren aus $\{0, 1\}^n$. KARP [25] hat 1972 gezeigt, dass das zugehörige Entscheidungsproblem NP-vollständig ist.

Man beachte, dass ein aus $L \geq 2$ Elementen aus X gebildeter Mehrheitsvektor nicht unbedingt X angehört. Um dennoch Mehrheitsbildung einsetzen zu können, bieten sich folgende Möglichkeiten an: Man kann einen Mehrheitsvektor $x \in \{0, 1\}^n \setminus X$ „reparieren“, indem man x durch ein geeignetes Element von X ersetzt. Das kann beispielsweise ein $x' \in X$ sein, welches x besonders „ähnlich“ ist, d.h. sich in minimaler Anzahl an Komponenten von x unterscheidet. Oder aber, man erweitert den Lösungsraum auf ganz $\{0, 1\}^n$. Hier muss durch entsprechende Neuwahl der Zielfunktion erreicht werden, dass wenigstens die lokal maximalen Lösungen in X liegen. Insbesondere terminiert lokale Suche dann stets in zulässigen Vektoren des $\{0, 1\}^n$. Für die vorliegende Arbeit wurde die letztgenannte Variante gewählt. So lassen sich Seiteneffekte aufgrund der speziellen Wahl von Reparaturmaßnahmen ausschließen. Zudem wird das Verhalten von Mehrheitsbildung beim Rucksack-Problem mit den Effekten bei den vorangegangenen Problemen vergleichbar. Sei also $f : \{0, 1\}^n \rightarrow \mathbb{R}$ definiert durch

$$f(x) := v(x) - p(x),$$

wobei

$$p(x) := \begin{cases} 0, & s(x) \leq S \\ C \cdot (s(x) - S), & s(x) > S \end{cases}$$

mit einer Konstante $C > \max\{v_i \mid i = 1, \dots, n\}$ sei. $p(x)$ ist ein **Strafterm**, der den Wert von $x \notin X$ je nach Höhe der Überschreitung der Rucksackgröße entsprechend mindert. Ist nun $x \in \{0, 1\}^n$ eine lokal maximale Lösung bezüglich der HAMMING-Nachbarschaft, so gilt (\star) , d.h. $x \in X$. Um dies einzusehen, sei $s(x) > S$, also $s(x) - S \geq 1$ angenommen. Wegen $S > 0$ existiert ein $i \in \{1, \dots, n\}$ mit $x_i = 1$. Definiere $y = (y_1, \dots, y_n) \in \{0, 1\}^n$ durch $y_i := 0$ und $y_j := x_j$ für jedes $j \in \{1, \dots, n\} \setminus \{i\}$. Falls $y \in X$, folgt

$$f(y) - f(x) = v(y) - v(x) + C(s(x) - S) = -v_i + \underbrace{C(s(x) - S)}_{> \max v_j} > 0$$

im Widerspruch zur lokalen Maximalität von x . Denselben Widerspruch erhält man, falls $y \notin X$:

$$f(y) - f(x) = v(y) - C(s(y) - S) - v(x) + C(s(x) - S) = -v_i + \underbrace{C s_i}_{> \max v_j} > 0$$

Der Lösungsraum $\{0, 1\}^n$ zusammen mit der HAMMING-Nachbarschaft und der Zielfunktion f ist also für das Rucksack-Problem unter Einsatz von Mehrheitsbildung geeignet. Man kann allerdings im Allgemeinen nicht erwarten, dass unter diesem Nachbarschaftsbegriff brauchbare lokal maximale Lösungen zustande kommen. Nach Erreichen der Schranke

S beim Bepacken des Rucksacks wird nicht berücksichtigt, ob durch Auspacken eines Pakets und Einpacken eines anderen ein besserer Wert erzielt werden kann. Die HAMMING-Nachbarschaft soll daher dahingehend erweitert werden, dass zu den HAMMING-Nachbarn eines Vektors $x = (x_1, \dots, x_n) \in \{0, 1\}^n$ alle Lösungen

$$y^{ij}(x) := (x_1, x_2, \dots, x_{i-1}, 1 - x_i, x_{i+1}, x_{i+2}, \dots, x_{j-1}, 1 - x_j, x_{j+1}, x_{j+2}, \dots, x_n)$$

mit $1 \leq i < j \leq n$ hinzukommen. Da die Nachbarn $y^{ij}(x)$ durch gleichzeitiges Flippen zweier Komponenten aus x hervorgehen, soll von der **Doppel-Flip-Nachbarschaft** gesprochen werden. Gilt speziell $x_i \neq x_j$, entsteht $y^{ij}(x)$ anschaulich durch Austausch des i -ten mit dem j -ten Paket aus x . Insgesamt hat jede Lösung $x \in \{0, 1\}^n$ genau $n + \binom{n}{2}$ Nachbarn. Natürlich ist jede lokal maximale Lösung bezüglich dieses erweiterten Nachbarschaftsbegriffs auch bezüglich der HAMMING-Nachbarschaft lokal maximal. Insbesondere bleibt Bedingung (\star) für alle lokal maximalen Lösungen gültig.

1.2.6 Das Counting-Ones-Problem

Das **Counting-Ones-Problem** ist aus Sicht der Optimierung trivial: Hier ist $n \geq 1$ und $f : \{0, 1\}^n \rightarrow \mathbb{R}$ definiert durch

$$f(x_1, \dots, x_n) := \#\{i : x_i = 1\}.$$

Die Zielfunktion f ist zu maximieren. Trivialerweise gilt $\max f = n$. Der Einsvektor $(1, 1, \dots, 1)$ ist die einzige global (bzw. lokal) maximale Lösung. Die einfache Struktur dieses „Problems“ wird für ein besseres Verständnis von speziellen Effekten der Mehrheitsbildung sorgen.

Kapitel II

Einstufige Mehrheitsbildung

In diesem Kapitel werden experimentelle Ergebnisse der Mehrheitsbildung für die in Abschnitt 1.2 genannten Optimierungsprobleme mit Lösungsraum $V = \{0, 1\}^n$ vorgestellt. Mehrheitsvektoren entstehen durch Mehrheitsbildung aus $L \geq 2$ lokal maximalen Lösungen $x^1, x^2, \dots, x^L \in V$ des Problems (vgl. Definition 1.1). x^1, \dots, x^L werden durch *zufällige Läufe* lokaler Suche ermittelt, d.h. durch wiederholte lokale Suche mit zufällig gewähltem Startvektor in jedem Lauf. An die L zufälligen Läufe schließt sich ein *Mehrheitslauf* an: hier wird als Startvektor ein Mehrheitsvektor x zu x^1, \dots, x^L gewählt. Die lokal maximale Lösung aus dem Mehrheitslauf wird dann hinsichtlich ihres Zielfunktionswerts mit den Lösungen x^1, \dots, x^L verglichen. Natürlich eignen sich auch lokal maximale Lösungen, die aus Mehrheitsläufen stammen, zur Gewinnung neuer Mehrheitsvektoren. Dies führt zu einem mehrstufigen, iterierten Einsatz des Mehrheitsprinzips. Das Augenmerk soll in diesem Kapitel jedoch zunächst auf einstufige Verfahren gerichtet sein.

2.1 Ausgewählte Max-SAT-Benchmarks

Seit 1990 finden am *Center for Discrete Mathematics and Theoretical Computer Science*, kurz DIMACS, der *Rutgers University* des Staates New Jersey regelmäßig Workshops mit ausgewählten Themen aus Diskreter Mathematik und Kombinatorischer Optimierung statt. Das Ziel besteht darin, weltweite Forschungsaktivitäten zu koordinieren und dabei besonderes Augenmerk auf praktische Lösungsansätze und Heuristiken für schwierige Probleme zu richten. Einer dieser Workshops mit dem Titel „NP Hard Problems: Maximum Clique, Graph Coloring, and Satisfiability“ [23] wurde unter Leitung von TRICK, CHVATAL, COOK, JOHNSON, MCGEOCH und TARJAN in den Jahren 1992/93 vorbereitet und vom 11. bis 13. Oktober 1993 abgehalten. Für die vorliegende Arbeit wurden einige der Max-SAT-Instanzen aus dem Benchmark-Katalog¹ für diesen Workshop willkürlich ausgewählt und auf das Abschneiden von Mehrheitsbildung hin untersucht.

2.1.1 Die Instanz „Hanoi 5“

Eines der wohl schwierigsten SAT-Benchmarks für den '93er-DIMACS-Workshop trug die Bezeichnung „Hanoi 5“ und kodierte eine Menge von Axiomen zur Beschreibung einer Instanz des bekannten Problems „Türme von Hanoi“ mit fünf Ringen. Zum damaligen Zeitpunkt wusste man zwar, dass „Hanoi 5“ lösbar ist, doch existierte bis dato kein Algorithmus, welcher eine Variablenbelegung zur Erfüllung aller Klauseln finden konnte.

¹via FTP erhältlich unter <ftp://dimacs.rutgers.edu/pub/challenge/sat/benchmarks/cnf/> bzw. <ftp://dimacs.rutgers.edu/pub/challenge/sat/contributed/UCSC/instances/cnf-bf/>

„Hanoi 5“ wird mit 1931 Variablen und 14468 Klauseln angegeben, wobei 365 Klauseln Länge 1, 7190 Klauseln Länge 2, 6820 Klauseln Länge 3, 31 Klauseln Länge 5 und 62 Klauseln Länge 8 haben.

Im Folgenden werden Ergebnisse aus je 2000 Läufen verschiedener Varianten wiederholter lokaler Suche vorgestellt.² Die 2000 Läufe unterteilten sich in 500 Blöcke zu je vier Läufen. Jeder Block bestand aus drei zufälligen Läufen und einem anschließenden Mehrheitslauf, dessen Startvektor durch Mehrheitsbildung aus den drei vorangegangenen lokal maximalen Lösungen gebildet wurde. Bei jeder getesteten Variante lokaler Suche wurden 2000 verschiedene lokal maximale Lösungen gefunden. Dies weist auf eine enorme Zahl aller existierenden lokal maximalen Lösungen hin.

1. Wiederholte lokale Suche ohne Sidesteps:

a) *Deterministisch wiederholte zyklische schnelle lokale Suche:*

Für die wiederholte zyklische schnelle lokale Suche wurde als Permutation π der 1931 Suchrichtungen die identische Permutation festgehalten (vgl. Abschnitt 1.1.3). Die HAMMING-Nachbarn einer aktuellen Belegung $x \in \{0, 1\}^{1931}$ wurden also stets in der Reihenfolge der kanonischen Enumeration $y^1(x), y^2(x), \dots, y^{1931}(x)$ untersucht (vgl. Definition 1.2). Abbildung 2.1 zeigt ein Säulendiagramm, in dem

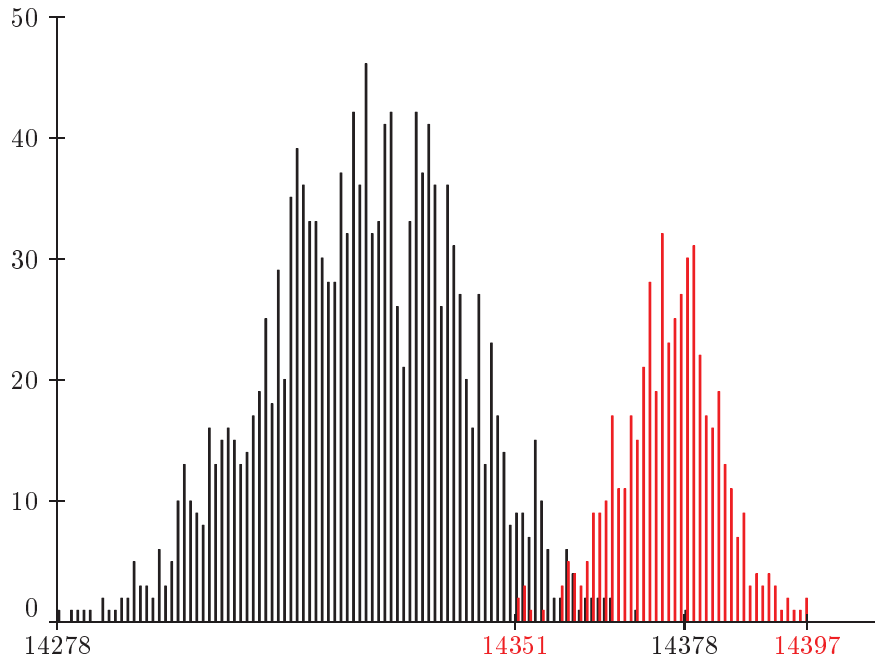


Abbildung 2.1: Häufigkeitsverteilung der gefundenen lokalen Maxima bei deterministisch wiederholter zyklischer schneller lokaler Suche

auf der Abszisse die f -Werte und auf der Ordinate ihre Häufigkeiten innerhalb der 2000 aufgetretenen lokal maximalen Lösungen aufgetragen sind. Die Summe dieser Häufigkeiten ist also 2000. Die Häufigkeiten der lokalen Maxima aus den Mehrheitsläufen sind rot, die aus den zufälligen Läufen schwarz gekennzeichnet.

²Um Verwechslungen mit Satzkommata zu vermeiden, werden Dezimalbrüche im Weiteren stets in der Punkt- statt Komma-Notation geschrieben.

Auffällig ist der deutliche Gütevorsprung der in den Mehrheitsläufen gewonnenen lokalen Maxima. Der beste erzielte f -Wert war 14397 und wurde in einem Mehrheitslauf erreicht. Als schlechtester Wert trat 14278 nach einem zufälligen Lauf auf. Der beste zufällige Lauf endete bei 14378, der schlechteste Mehrheitslauf bei 14351. Im Durchschnitt ergaben die lokal maximalen Lösungen aus den zufälligen Läufen einen f -Wert von 14326.5, aus den Mehrheitsläufen von 14375.2.

b) *Wiederholte volle lokale Suche:*

Unter den gleichen Bedingungen wie in a) fand nun ein Vergleich zwischen zufälligen Läufen und Mehrheitsläufen wiederholter voller lokaler Suche statt. Der beste Funktionswert aus einem Mehrheitslauf war 14405, der schlechteste 14353 (Abbildung 2.2). Aus den zufälligen Läufen traten f -Werte zwischen 14357 und 14241 auf. Im Mittel lagen die f -Werte nach den zufälligen Läufen voller Suche bei 14304.3, nach den Mehrheitsläufen bei 14377.7.

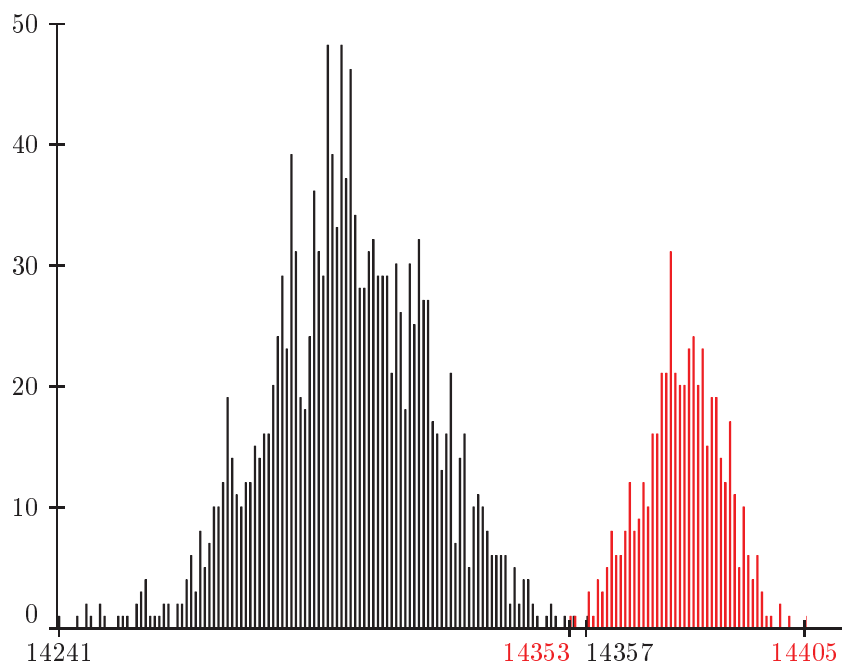


Abbildung 2.2: Häufigkeitsverteilung der gefundenen lokalen Maxima bei wiederholter voller lokaler Suche

2. *Wiederholte lokale Suche mit Sidesteps:*

Den beiden obigen Säulendiagrammen werden nun die Ergebnisse aus wiederholter lokaler Suche mit Erlaubnis von Sidesteps gegenüber gestellt. Es wurde die in Abschnitt 1.1.5 beschriebene Schrankenmethode verwendet. Zwischen je zwei Verbesserungsschritten innerhalb eines Laufs lokaler Suche durften höchstens 1931 Sidesteps vollzogen werden, also so viele Sidesteps, wie Variable vorkommen.

a) *Deterministisch wiederholte zyklische schnelle lokale Suche mit Sidesteps:*

Im Vergleich zur deterministisch wiederholten zyklischen schnellen lokalen Suche ohne Sidesteps traten wesentlich höhere und weniger verschiedene Zielfunktionswerte auf. Die nach den zufälligen Läufen aufgetretenen f -Werte lagen im Bereich von 14423 bis 14452, nach den Mehrheitsläufen zwischen 14435 und 14458. Im

Durchschnittlich traten nach den zufälligen Läufen f -Werte von 14440.8, nach den Mehrheitsläufen von 14448.4 auf.

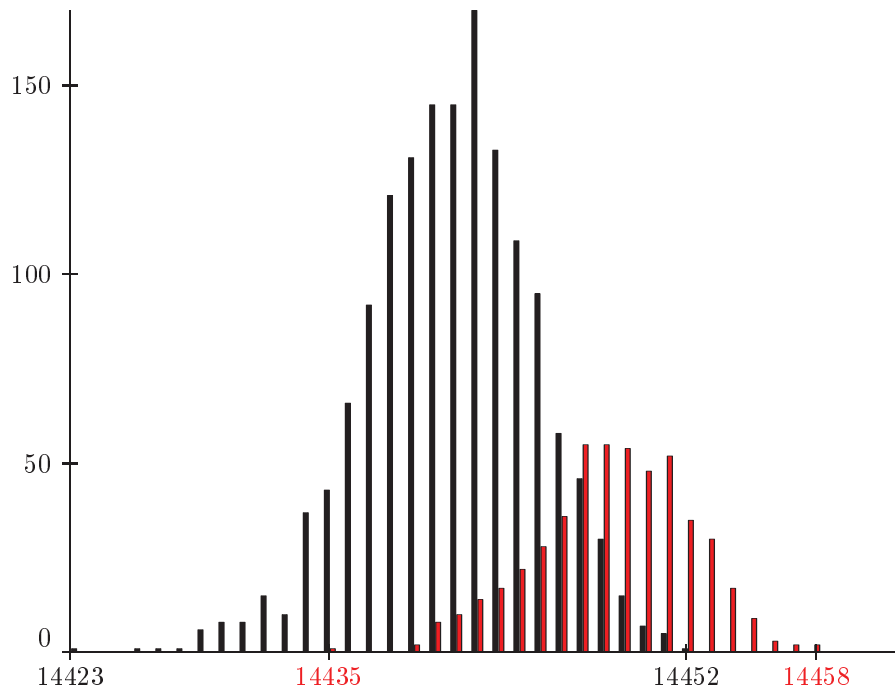


Abbildung 2.3: Häufigkeitsverteilung der gefundenen lokalen Maxima bei deterministisch wiederholter zyklischer schneller lokaler Suche mit Sidesteps

b) Wiederholte volle lokale Suche mit Sidesteps:

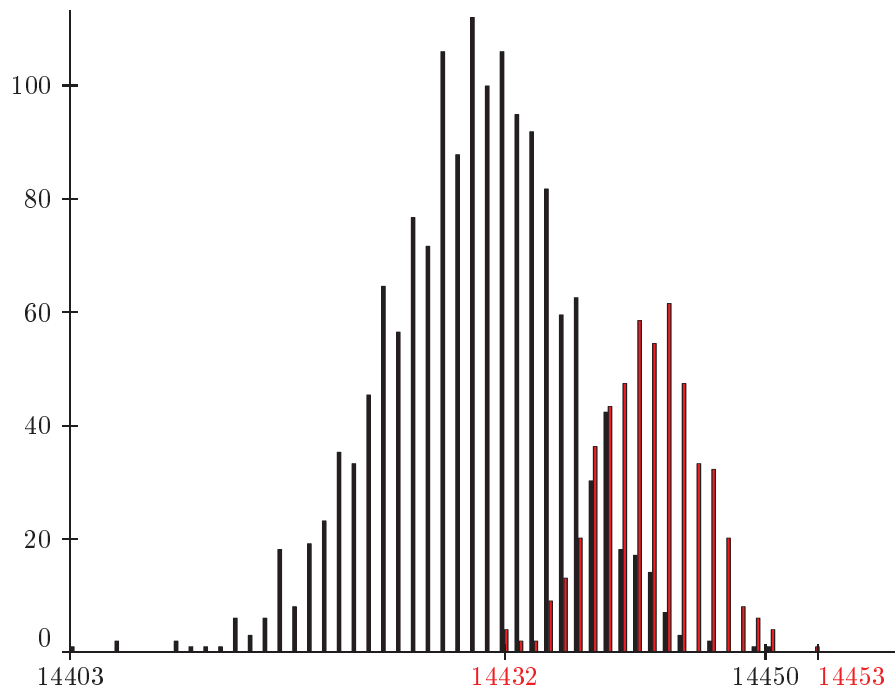


Abbildung 2.4: Häufigkeitsverteilung der gefundenen lokalen Maxima bei wiederholter voller lokaler Suche mit Sidesteps

Bei der wiederholten vollen lokalen Suche mit Sidesteps erbrachten die zufälligen Läufe f -Werte zwischen 14403 und 14450, im Durchschnitt 14430.0. Die f -Werte der Mehrheitsläufe lagen zwischen 14432 und 14453, im Mittel bei 14441.8.

Analoge Versuche wurden auch für die probabilistisch wiederholte zyklische schnelle lokale Suche ohne und mit Sidesteps angestellt. Auf die entsprechenden Säulendiagramme wird aus Platzgründen verzichtet. In Tabelle 2.1 sind alle wichtigen Ergebnisse bei den sechs Varianten lokaler Suche zusammengefasst. „dZS“ bzw. „pZS“ sind die Kurzformen für „deterministisch wiederholt, zyklisch schnell“ bzw. „probabilistisch wiederholt, zyklisch schnell“. „V“ steht für „voll“. Der Anhang „s“ bezeichnet die jeweilige Variante lokaler Suche mit Sidesteps. \max_{zf} bzw. \max_{Mh} bezeichnet den besten gefundenen f -Wert aus den 1500 zufälligen Läufen bzw. den 500 Mehrheitsläufen. \bar{f}_{zf} bzw. \bar{f}_{Mh} kürzt den jeweiligen durchschnittlichen f -Wert der aufgetretenen lokal maximalen Lösungen ab, gemittelt über die zugehörige Anzahl der Läufe. Die durchschnittlichen Schrittzahlen, also Übergänge zu HAMMING-Nachbarn inklusive Sidesteps während eines Laufs, sind als \bar{S}_{zf} bzw. \bar{S}_{Mh} angegeben. Mit \bar{F}_{zf} bzw. \bar{F}_{Mh} werden die durchschnittlichen Fragenzahlen bezeichnet.

	\max_{zf}	\max_{Mh}	\bar{f}_{zf}	\bar{f}_{Mh}	\bar{S}_{zf}	\bar{S}_{Mh}	\bar{F}_{zf}	\bar{F}_{Mh}
dZS	14378	14397	14326.5	14375.2	512.2	76.8	12796.6	6464.1
dZSs	14452	14458	14440.8	14448.4	4328.8	3454.9	290278.4	300434.4
pZS	14352	14390	14301.4	14366.7	504.7	83.6	10421.0	5871.1
pZSs	14451	14452	14437.5	14444.1	4008.9	3241.3	272495.8	278349.8
V	14357	14405	14304.3	14377.7	418.2	75.7	809471.3	148088.4
Vs	14450	14453	14430.0	14441.8	4238.1	3422.8	8185726.6	6611288.3

Tabelle 2.1: Ergebnisse aus zufälligen Läufen und Mehrheitsläufen lokaler Suche verschiedener Varianten bei „Hanoi 5“

Für das Benchmark-Problem „Hanoi 5“ lassen sich zusammenfassend bei allen getesteten Varianten lokaler Suche folgende Beobachtungen machen:

1. Mehrheitsläufe erzielen im Durchschnitt bessere Zielfunktionswerte als zufällige Läufe. Die besten gefundenen f -Werte stammen typischerweise aus Mehrheitsläufen.
2. Mehrheitsläufe machen im Durchschnitt weniger Schritte als zufällige Läufe. Dies ist ein Indiz dafür, dass Mehrheitsvektoren im Mittel einen geringeren HAMMING-Abstand zu lokal maximalen Lösungen haben als zufällige Startvektoren (vgl. S. 16).
3. Mehrheitsläufe stellen im Durchschnitt weniger Fragen als zufällige Läufe (jeweils ohne Sidesteps). Dies führt zu kürzeren Rechenzeiten.

2.1.2 Weitere Max-SAT-Benchmarks

Neben „Hanoi 5“ wurde eine Reihe anderer Instanzen des DIMACS-Katalogs ausgewählt. Unter den Original-Bezeichnungen der zugehörigen Datenfiles waren dies die Benchmarks bf0432-005, bf0432-011, bf1355-075, bf2670-001, hanoi4, par32-1, par32-2, g125_17, g125_18, g250_15 und g250_29. Diese Dateien enthalten Klauselmengen, welche

Instanz	L.S.	\max_{zf}	\max_{Mh}	\overline{f}_{zf}	\overline{f}_{Mh}	\overline{S}_{zf}	\overline{S}_{Mh}	\overline{F}_{zf}	\overline{F}_{Mh}
bf0432-005 $n = 1040$ $m = 3668$	V	3609	3617	3578.5	3595.8	210.4	41.9	278440.1	56549.3
	S	3600	3611	3570.5	3589.8	244.4	46.2	8495.9	5970.7
	pZS	3595	3609	3570.5	3589.1	242.9	46.3	4295.9	3337.5
bf0432-011 $n = 1053$ $m = 3743$	V	3678	3689	3650.7	3668.5	212.4	42.8	281040.8	57684.6
	S	3675	3683	3642.4	3661.8	247.1	47.7	8549.2	5982.8
	pZS	3676	3686	3642.6	3662.1	246.1	47.0	4316.4	3379.4
bf1355-075 $n = 2180$ $m = 6778$	V	6619	6644	6578.7	6607.0	495.3	94.2	1866040.4	358094.9
	S	6607	6635	6559.0	6595.6	575.4	105.6	28384.9	20873.6
	pZS	6602	6622	6559.5	6595.7	574.9	105.2	13481.5	10775.2
bf2670-001 $n = 1393$ $m = 3434$	V	3341	3352	3311.2	3326.6	336.5	60.6	1637339.7	298854.1
	S	3333	3342	3296.8	3313.4	386.5	65.9	32524.4	22976.1
	pZS	3329	3348	3296.7	3314.4	385.7	65.6	14739.4	12127.0
hanoi4 $n = 718$ $m = 4934$	V	4897	4910	4871.3	4896.0	148.7	30.6	107459.2	22704.6
	S	4894	4907	4868.2	4891.2	182.0	34.2	5002.8	2860.2
	pZS	4899	4906	4869.0	4892.3	181.0	34.1	2924.8	1850.6
par32-1 $n = 3176$ $m = 10277$	V	9773	9803	9716.2	9760.5	533.0	241.7	1696047.5	770719.9
	S	9777	9800	9709.7	9751.0	617.1	261.0	22767.7	19562.6
	pZS	9768	9801	9709.9	9751.8	614.5	259.0	11600.5	10020.8
par32-2 $n = 3176$ $m = 10253$	V	9759	9791	9691.9	9737.2	533.0	242.4	1695859.1	772936.8
	S	9754	9780	9686.5	9729.2	616.3	259.1	22587.2	19600.9
	pZS	9739	9778	9685.1	9727.8	613.9	258.5	11585.6	10015.6
g125_17 $n = 2125$ $m = 66272$	V	66243	66247	66234.3	66239.3	70.8	80.6	152568.0	173476.5
	S	66243	66247	66234.7	66239.4	72.3	80.6	8416.1	7562.8
	pZS	66246	66247	66234.8	66239.2	72.6	80.6	4928.4	4132.9
g125_18 $n = 2250$ $m = 70163$	V	70138	70142	70128.7	70134.0	73.5	83.6	167571.0	190341.0
	S	70139	70142	70129.2	70133.9	75.4	83.6	8975.5	8025.6
	pZS	70139	70143	70129.2	70134.1	75.3	83.9	5252.0	4371.7
g250_15 $n = 3750$ $m = 233965$	V	233838	233840	233820.0	233825.2	89.2	100.6	338090.0	380962.5
	S	233837	233840	233820.3	233825.4	90.4	100.8	15486.8	14434.9
	pZS	233839	233841	233820.5	233824.9	90.5	100.5	8610.5	7334.6
g250_29 $n = 7250$ $m = 454622$	V	454562	454567	454547.4	454556.0	147.1	170.2	1073860.3	1241098.5
	S	454562	454570	454548.4	454556.0	150.2	170.2	33095.9	29661.7
	pZS	454564	454566	454548.3	454555.7	150.1	169.4	17720.7	14253.6

Tabelle 2.2: *Ergebnisse aus zufälligen Läufen und Mehrheitsläufen lokaler Suche bei verschiedenen Max-SAT-Instanzen*

Probleminstanzen aus den Bereichen der Fehleranalyse von Schaltkreisen (Präfix „bf“), Türme von Hanoi (Präfix „hanoi“), Parameteridentifikation gewisser BOOLEscher Funktionen (Präfix „par“) sowie der Färbbarkeit von Graphen (Präfix „g“) kodieren. Den beigefügten Dokumentationen zufolge sind bei den Hanoi-, Parameteridentifikations- und Graphenfärbbarkeits-Benchmarks jeweils alle Klauseln erfüllbar. Bei den Schaltkreis-Beispielen werden keine Angaben zur Erfüllbarkeit gemacht.

Tabelle 2.2 zeigt die Ergebnisse des Einsatzes von Mehrheitsbildung unter verschiedenen Varianten lokaler Suche. Es wurden pro Instanz und Suchvariante 2000 Läufe gemacht, und zwar wie schon bei „Hanoi 5“ jeweils 500 Blöcke zu je drei zufälligen Läufen und einem anschließenden Mehrheitslauf. Der Startvektor eines Mehrheitslaufs ging jeweils durch Mehrheitsbildung aus den drei lokal maximalen Lösungen der vorangegangenen zufälligen Läufe hervor. Zum Einsatz kamen wiederholte volle lokale Suche („V“), wiederholte schnelle lokale Suche („S“) und probabilistisch wiederholte zyklische schnelle lokale Suche („pZS“). In der ersten Spalte sind außer dem Dateinamen der Instanz die Zahl n der Variablen und die Zahl m der Klauseln angegeben. Die Bedeutung der weiteren Spalten ist wie in Tabelle 2.1.

Unter allen Instanzen und Suchvarianten wurden jeweils 2000 verschiedene lokal maximale Lösungen gefunden. Bei den Schaltkreis-, Hanoi- und Parameteridentifikations-Beispielen traten ähnlich gute Erfolge der Mehrheitsbildung auf. Die Graphenbeispiele unterschieden sich von diesen in zweierlei Hinsicht: Zum einen waren die Differenzen bei den f -Werten nicht allzu groß. Man beachte aber hierzu, dass schon die zufällige lokale Suche f -Werte mit einer Approximationsgüte von mehr als 99.9% zum globalen Optimum m lieferte. Zum anderen traten in den Mehrheitsläufen sowohl hinsichtlich der Schrittzahlen als auch teilweise der Fragenzahlen höhere Durchschnittswerte als in den zufälligen Läufen auf. Ungeachtet dessen besteht dennoch kein gewichtiger Grund, vom Einsatz der Mehrheitsbildung abzuraten.

2.2 Erfolgsquoten bei zufällig erzeugten Instanzen

Die Instanzen aller in Abschnitt 1.2 genannten Optimierungsprobleme hängen von bestimmten numerischen Parametern ab. Bei Max- k -SAT sind dies etwa die Zahl n der Variablen, die Zahl m der Klauseln und die Zahl k der Literale pro Klausel. Am Beispiel des Ising-Spinalgas-Problems auf dem $(n \times n)$ -Torus lässt sich n als Problemparameter auffassen. Wenngleich die Menge aller Probleminstanzen unendlich groß ist, existieren zu jedem festen Parameter bzw. Parametersatz (bei mehreren Parametern) r nur endlich viele Instanzen. Die Menge dieser Instanzen wird mit $I(r)$ bezeichnet.

In diesem Abschnitt soll der „Erfolg“ der Mehrheitsbildung in Abhängigkeit eines Parametersatzes r dargestellt werden. Hierzu ist ein numerischer „Messwert“ des Erfolgs erforderlich. Falls $f > 0$, ist es beispielsweise denkbar, den Mittelwert aller Quotienten der Form

$$\frac{f(x)}{\max\{f(x^1), f(x^2), \dots, f(x^L)\}}$$

mit $L \geq 2$ zu bilden. $x^1, x^2, \dots, x^L \in \{0, 1\}^n$ seien dabei lokal maximale Lösungen und $x \in \{0, 1\}^n$ ein zugehöriger Mehrheitsvektor. Man beachte hierzu, dass die Optimierung in dem hier behandelten Sinne letztlich auf die Ermittlung einer einzelnen Lösung mit möglichst großem f -Wert abzielt. Dies rechtfertigt die Maximum-Bildung im Nenner. Gemittelt wird über alle solchen x^1, \dots, x^L und x und alle Instanzen aus $I(r)$. In praktischen Versuchen stellte sich heraus, dass $f(x)$ typischerweise schlechter als der Durchschnitt von

$f(x^1), \dots, f(x^L)$ ist. Stattdessen soll der Erfolg der Mehrheitsbildung an der Antwort auf folgende Fragestellung gemessen werden:

Bei gegebener Probleminstanz erlaube die Rechenzeit (mindestens) $L + 1$ Läufe lokaler Suche einer bestimmten Variante. Welche der folgenden Alternativen ist vorzuziehen?

- Ermittlung von $L + 1$ lokal maximalen Lösungen x^1, x^2, \dots, x^{L+1} aus $L + 1$ unabhängigen zufälligen Läufen
- Ermittlung von L lokal maximalen Lösungen y^1, y^2, \dots, y^L aus L unabhängigen zufälligen Läufen und danach einer lokal maximalen Lösung y^{L+1} aus einem Mehrheitslauf

Für letztere Möglichkeit soll die Startlösung von Lauf $L + 1$ durch Mehrheitsbildung aus y^1, \dots, y^L entstehen. Zum Vergleich stehen $\max_{\text{zf}} := \max \{f(x^1), \dots, f(x^{L+1})\}$ und $\max_{\text{Mh}} := \max \{f(y^1), \dots, f(y^{L+1})\}$. Welcher Wert ist typischerweise größer?

Zur experimentellen Untersuchung dieser Frage wird eine (möglichst) große Zahl B von unabhängig und zufällig gewählten Instanzen aus $I(r)$ gemäß der diskreten Gleichverteilung auf $I(r)$ erzeugt. Zu jeder Instanz werden mittels wiederholter lokaler Suche einer Variante \mathcal{S} lokal maximale Lösungen x^1, \dots, x^{L+1} und y^1, \dots, y^{L+1} wie oben ermittelt. Dabei wird festgehalten, wie oft \max_{Mh} größer, gleich groß oder kleiner als \max_{zf} ist. Die zugehörigen Anzahlen seien $\#\{\max_{\text{Mh}} > \max_{\text{zf}}\}$, $\#\{\max_{\text{Mh}} = \max_{\text{zf}}\}$ und $\#\{\max_{\text{Mh}} < \max_{\text{zf}}\}$.

DEFINITION 2.1 (Erfolgsquote)

Unter dem oben beschriebenen Szenario heißt

$$\frac{1 \cdot \#\{\max_{\text{Mh}} > \max_{\text{zf}}\} + \frac{1}{2} \cdot \#\{\max_{\text{Mh}} = \max_{\text{zf}}\} + 0 \cdot \#\{\max_{\text{Mh}} < \max_{\text{zf}}\}}{B}$$

die (empirische) **Erfolgsquote** von Mehrheitsbildung unter r , L und \mathcal{S} .

Per Definition ist die Erfolgsquote eine rationale Zahl aus dem Intervall $[0, 1]$. Die Erwartung besteht darin, dass die Erfolgsquote unter r , L und \mathcal{S} für alle oder wenigstens viele bzw. in irgendeiner Weise relevante Problemgrößen r größer als $\frac{1}{2}$ ist. In diesem Fall kann man den Einsatz der Mehrheitsbildung unter einer beliebigen Instanz aus $I(r)$ als ratsam ansehen, wenn keine zusätzlichen Informationen über die Struktur der Instanz vorliegen.

Man beachte, dass die Erfolgsquoten nicht nur von den Größen r , L und \mathcal{S} abhängen. Sie werden vielmehr auch durch praktische Faktoren wie die Beispielszahl B oder die verwendeten Pseudozufallszahlen während des Ablaufs des Programms maßgeblich beeinflusst. Durch Übergang zu dem folgenden stochastischen Modell kann man sich von diesen Seiteneffekten lösen: Sei $\Omega := I(r) \times (\{0, 1\})^{2 \cdot (L+1)}$. Weiter sei $P : 2^\Omega \rightarrow [0, 1]$ das Wahrscheinlichkeitsmaß, das jedem Elementarereignis $\{(I, x^1, \dots, x^{L+1}, y^1, \dots, y^{L+1})\} \in 2^\Omega$ die Wahrscheinlichkeit zuordnet, mit der $I \in I(r)$ gewählt wird und $x^1, \dots, x^{L+1}, y^1, \dots, y^{L+1}$ unter Instanz I und Variante \mathcal{S} lokaler Suche als lokal maximale Lösungen sowie y^{L+1} als Mehrheitsvektor zu y^1, \dots, y^L auftreten. Schließlich sei die Zufallsvariable $X_{r,L,\mathcal{S}} : (\Omega, P) \rightarrow \mathbb{R}$ definiert durch

$$X_{r,L,\mathcal{S}}(\omega) := \begin{cases} 1, & \text{falls } \max_{\text{Mh}} > \max_{\text{zf}} \\ \frac{1}{2}, & \text{falls } \max_{\text{Mh}} = \max_{\text{zf}} \\ 0, & \text{falls } \max_{\text{Mh}} < \max_{\text{zf}} \end{cases}$$

für alle $\omega = (I, x^1, \dots, x^{L+1}, y^1, \dots, y^{L+1}) \in \Omega$. Dabei sei f die durch I gegebene Zielfunktion, $\max_{\text{zf}} := \max \{f(x^1), \dots, f(x^{L+1})\}$ und $\max_{\text{Mh}} := \max \{f(y^1), \dots, f(y^{L+1})\}$. Nach dem Gesetz der großen Zahlen bildet die empirische Erfolgsquote von Mehrheitsbildung unter r , L und \mathcal{S} eine Näherung für den Erwartungswert von $X_{r,L,\mathcal{S}}$.

DEFINITION 2.2 (Erfolg)

Die Erwartung von $X_{r,L,\mathcal{S}}$ heißt der **Erfolg** von Mehrheitsbildung unter r , L und \mathcal{S} .

Da die theoretische Behandlung des Erfolgs von Mehrheitsbildung zu komplex ist, beschränken sich die Untersuchungen der vorliegenden Arbeit auf die Ermittlung empirischer Erfolgsquoten. Beobachtungen für empirische Erfolgsquoten werden als Vermutungen für den Erfolg von Mehrheitsbildung formuliert.

Als weiteres Vergleichskriterium zwischen zufälligen Läufen und Mehrheitsläufen lokaler Suche werden ihre Schrittzahlen herangezogen. Diese geben die Anzahl der Übergänge zu besseren Nachbarlösungen während eines Laufs wieder (vgl. S. 16). Bei fest gewähltem Parametersatz r , fester Gruppengröße L und fester Variante \mathcal{S} lokaler Suche werden die auftretenden Schrittzahlen über alle entsprechenden Mehrheitsläufe und alle erzeugten Instanzen aus $I(r)$ gemittelt. Dieser experimentell entstehende Zahlenwert sei $S_{\text{Mh}}(r, L, \mathcal{S})$. Analog werden die durchschnittlichen Schrittzahlen $S_{\text{zf}}(r, \mathcal{S})$ zufälliger Läufe gebildet. Von besonderem Interesse sind dann die durchschnittlichen Schrittzahlen von Mehrheitsläufen relativ zu den durchschnittlichen Schrittzahlen zufälliger Läufe:

DEFINITION 2.3 (Schrittzahlquotient)

Der Quotient $S_{\text{Mh}}(r, L, \mathcal{S})/S_{\text{zf}}(r, \mathcal{S})$ heißt (empirischer) **Schrittzahlquotient** von Mehrheitsbildung unter r , L und \mathcal{S} .

O.B.d.A. existieren zu jedem Parametersatz r einige Instanzen, deren Lösungen nicht allesamt lokal maximal sind. Dann gehen bei hinreichend großer Anzahl gerechneter Instanzen und zufälliger Läufe positive Schrittzahlen in die Mittelung ein, so dass $S_{\text{zf}}(r, \mathcal{S}) > 0$ und obiger Quotient wohldefiniert ist.

Hängen die Instanzen des Problems nur von einem einzigen Parameter ab, lässt sich die Folge der Erfolgsquoten in Abhängigkeit dieses Parameters in einer zweidimensionalen Graphik veranschaulichen. In diesem Fall werden aus optischen Gründen aufeinanderfolgende Datenpunkte stets durch eine Linie verbunden. Die graphische Veranschaulichung einer Folge von (empirischen) Erfolgsquoten in Abhängigkeit eines Parameters r auf diese Weise sei als **Erfolgskurve** bezeichnet. Analog sei unter der **Kurve der Schrittzahlquotienten** in Abhängigkeit des Parameters r die graphisch durch Linien verbundene Folge der (empirischen) Schrittzahlquotienten in r zu verstehen.

2.2.1 Erfolgsquoten für Max-3-SAT

Eine Max-3-SAT-Instanz basiert auf den Problemparametern n und m mit $n, m \geq 1$. Dabei bezeichnet n die Zahl der BOOLEschen Variablen X_1, \dots, X_n und m die Anzahl der Klauseln. Jede Klausel ist Disjunktion von 3 Literalen. Zu jedem festen Parametersatz

$r = (n, m)$ wird eine Instanz aus $I(r)$ wie folgt zufällig bestimmt: Für jedes der $3m$ Literale wird je mit Wahrscheinlichkeit $\frac{1}{n}$ eine der n Variablen gewählt und diese anschließend mit Wahrscheinlichkeit $\frac{1}{2}$ negiert. Auf diese Weise tritt jede der insgesamt $(2n)^{3m}$ möglichen Instanzen mit derselben Wahrscheinlichkeit auf.

Im Folgenden werden die Erfolgsquoten der Mehrheitsbildung unter ausgewählten Parametersätzen $r = (n, m)$ in Abhängigkeit eines der Parameter n bzw. m dargestellt. Variiert man etwa n , so ist intuitiv bereits klar, dass m für sinnvolle Untersuchungen mit n korreliert sein sollte. Sehr wenige Klauseln im Vergleich zur Variablenzahl n sind nämlich typischerweise problemlos zu erfüllen, so dass sowohl die zufälligen Läufe als auch die Mehrheitsläufe sicher das globale Maximum erreichen. Unter jedem festen m tritt dieser Fall bei wachsendem n irgendwann ein. Interessant sind Parametersätze, die für nicht-triviale Instanzen sorgen. SELMAN, LEVESQUE und MITCHELL [27] haben beobachtet, dass es zu jeder Variablenzahl n eine eindeutig bestimmte Klauselanzahl $m(n)$ mit folgenden Eigenschaften gibt: Für zufällig gewählte 3-SAT-Instanzen zu den Parametern n und $m(n)$ mit paarweise verschiedenen Variablen innerhalb einer Klausel ist es „besonders schwierig“ zu entscheiden, ob alle Klauseln zugleich erfüllbar sind. Als Maß für die Schwierigkeit haben sie die Zahl der rekursiven Aufrufe während des Ablaufs des Resolutionsverfahrens von DAVIS und PUTNAM [11] gewählt. Ferner schien die Zahl $m(n)$ in einem Bereich zu liegen, in dem es mit Wahrscheinlichkeit $\frac{1}{2}$ eine Belegung gibt, die alle Klauseln erfüllt. Im Einzelnen wurden folgende Werte beobachtet:

$$\begin{aligned} n = 20 : \quad m(n) &\approx 4.55 n \\ n = 50 : \quad m(n) &\approx 4.31 n \\ n = 140 : \quad m(n) &\approx 4.30 n \end{aligned}$$

$$\text{Vermutung in [27] :} \quad \lim_{n \rightarrow \infty} \frac{m(n)}{n} = 4.25$$

Für die folgenden Untersuchungen wird daher zu jeder Variablenzahl n die Klauselzahl

$$m^* := m^*(n) := \lceil 4.25n \rceil$$

verwendet. Diese Festlegung erscheint — zumindest für erste Versuche — sinnvoll, auch wenn sich im vorliegenden Modell Variable innerhalb einer Klausel wiederholen dürfen. Abbildung 2.5 zeigt die Erfolgskurve unter $r = (n, m^*(n))$, $L = 3$ und wiederholter voller lokaler Suche bzw. probabilistisch wiederholter zyklischer schneller lokaler Suche. Da die Läufe voller lokaler Suche wesentlich mehr Rechenzeit als die Läufe zyklischer schneller lokaler Suche benötigen, wurde dort auf größere n als 2000 verzichtet. Zudem wurden Erfolgsquoten nicht für *alle* n aus $\{1, \dots, 2000\}$ bzw. $\{1, \dots, 5000\}$ berechnet, sondern nur für $n \in \{1, 2, \dots, 100\} \cup \{200, 300, 400, \dots, 2000\}$ bei wiederholter voller lokaler Suche und nur für $n \in \{1, 2, \dots, 100\} \cup \{105, 110, 115, \dots, 300\} \cup \{350, 400, 450, \dots, 3000\} \cup \{3100, 3200, 3300, \dots, 5000\}$ bei wiederholter zyklischer schneller lokaler Suche. Unter beiden Varianten wurde über 100 000 Beispiel-Instanzen für jedes $n \leq 100$ beziehungsweise über 10 000 Instanzen für jedes $n > 100$ gemittelt.

Unabhängig von der Strategie lokaler Suche übertrifft die Mehrheitsbildung die Erwartung, dass ihre Erfolgsquoten unter obigen Parametern für fast alle n über der $\frac{1}{2}$ -Marke

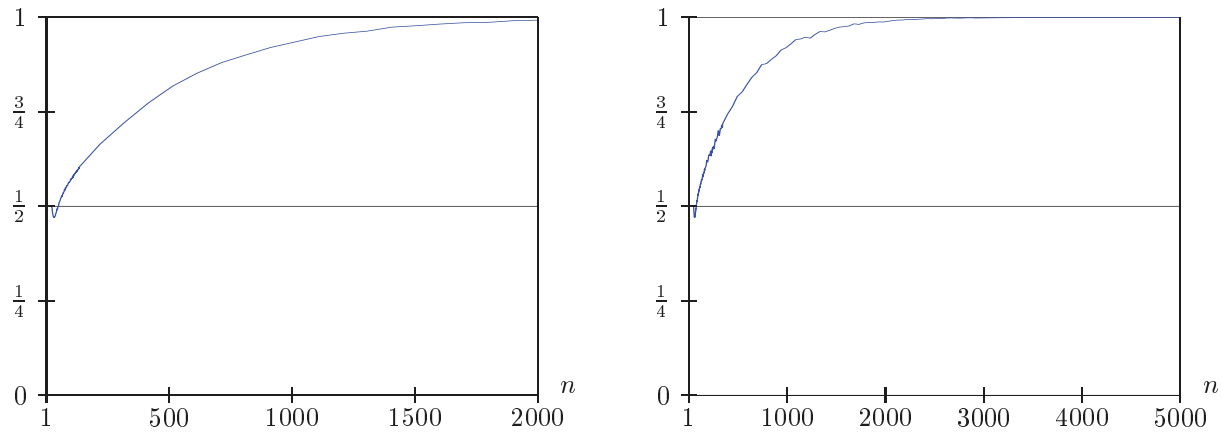


Abbildung 2.5: **(Max-3-SAT)** Erfolgskurven unter $r = (n, m^*)$, $L = 3$ und wiederholter voller lokaler Suche (links) bzw. probabilistisch wiederholter zyklischer schneller lokaler Suche (rechts) in Abhängigkeit von n

liegen, bei weitem: Die Folgen der Erfolgsquoten konvergieren unter beiden Varianten lokaler Suche für $n \rightarrow \infty$ gegen 1. Dass die beiden Erfolgskurven in $\frac{1}{2}$ „starten“, ist klar, denn für $n = 1$ findet jeder Lauf lokaler Suche unter beliebiger Suchstrategie trivialerweise das globale Maximum. Ungeklärt bleibt an dieser Stelle, was es mit dem „kurzzeitigen“ geringen Abfall der Erfolgskurven unter $\frac{1}{2}$ auf sich hat, bevor die Folgen gegen 1 konvergieren. Die Einschränkung einer Erfolgskurve auf ein Intervall kleiner Parameter mit Erfolgsquoten von höchstens $\frac{1}{2}$ wird im Folgenden **Misserfolgstal** genannt. Eine Diskussion zum Misserfolgstal wird in Abschnitt 2.3 geführt.

Ein Vergleich von $L = 3$ mit anderen Gruppengrößen L zeigt, dass bei Max-3-SAT sowohl die Konvergenz der Erfolgsquotenfolge in n gegen 1 als auch das Misserfolgstal typisch sind. Die Abbildungen 2.6 und 2.7 veranschaulichen die Folgen der Erfolgsquoten unter $r = (n, m^*)$ und $L = 2, 3, \dots, 9$ für wiederholte volle lokale Suche und probabilistisch wiederholte zyklische schnelle lokale Suche, jeweils im Bereich $1 \leq n \leq 100$. Für jede Variablenzahl n wurden die entsprechenden Erfolgsquoten durch Auswertung über 100 000 i.i.d. gewählte Probleminstanzen ermittelt. Es fällt sofort auf, dass alle Erfolgskurven für kleine n ein Misserfolgstal aufweisen, welches mit wachsender Gruppengröße L flacher und kürzer wird. Letztere Beobachtung sollte nicht weiter verwundern, da mit zunehmendem L dem einen Mehrheitslauf innerhalb einer Gruppe von $2 \cdot (L + 1)$ Läufen lokaler Suche immer weniger Bedeutung zukommt. Auch nach Überschreiten der $\frac{1}{2}$ -Marke gilt für jedes n : je größer L , desto größer die Erfolgsquote. Anhand der Graphiken ist dies für $L \geq 5$ zwar nicht einwandfrei zu erkennen, erscheint aber zumindest als glaubhaft. Die beiden Varianten der wiederholten lokalen Suche an sich haben auf das Abschneiden der Mehrheitsbildung kaum Einfluss. Diese Beobachtungen führen zu folgender verallgemeinerten

VERMUTUNG 2.4

Der Erfolg von Mehrheitsbildung für das Problem Max-3-SAT in Abhängigkeit von $r = (n, m^(n))$ konvergiert unter jeder Gruppengröße $L \geq 2$ und beliebiger in Abschnitt 1.1 genannter Variante wiederholter lokaler Suche für $n \rightarrow \infty$ gegen 1.*

Vergleicht man die Abbildungen 2.6 und 2.7 etwas genauer, fallen am rechten Rand, d.h.

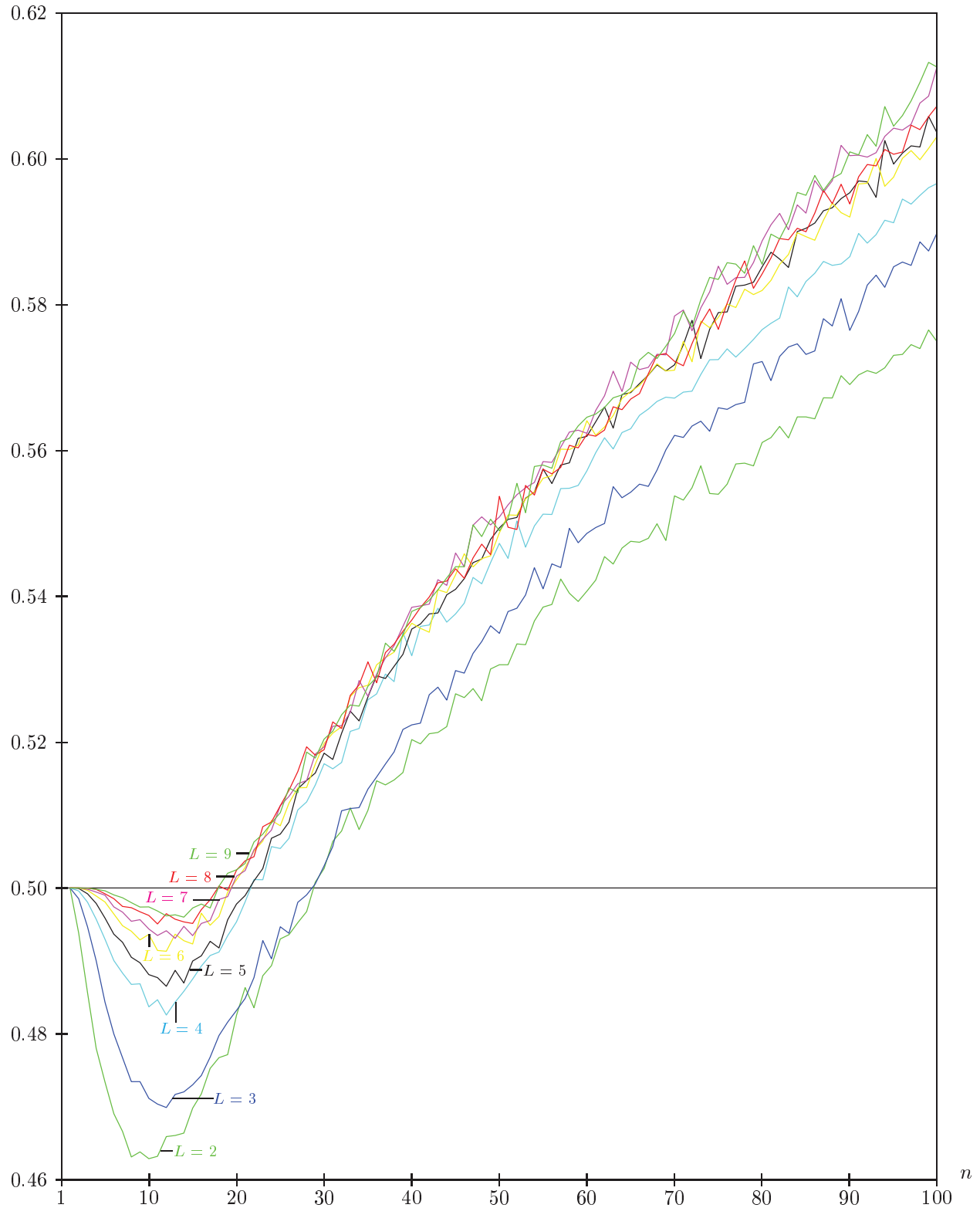


Abbildung 2.6: (**Max-3-SAT**) Erfolgskurven unter $r = (n, m^*)$, verschiedenen Gruppengrößen L und wiederholter voller lokaler Suche in Abhängigkeit von n

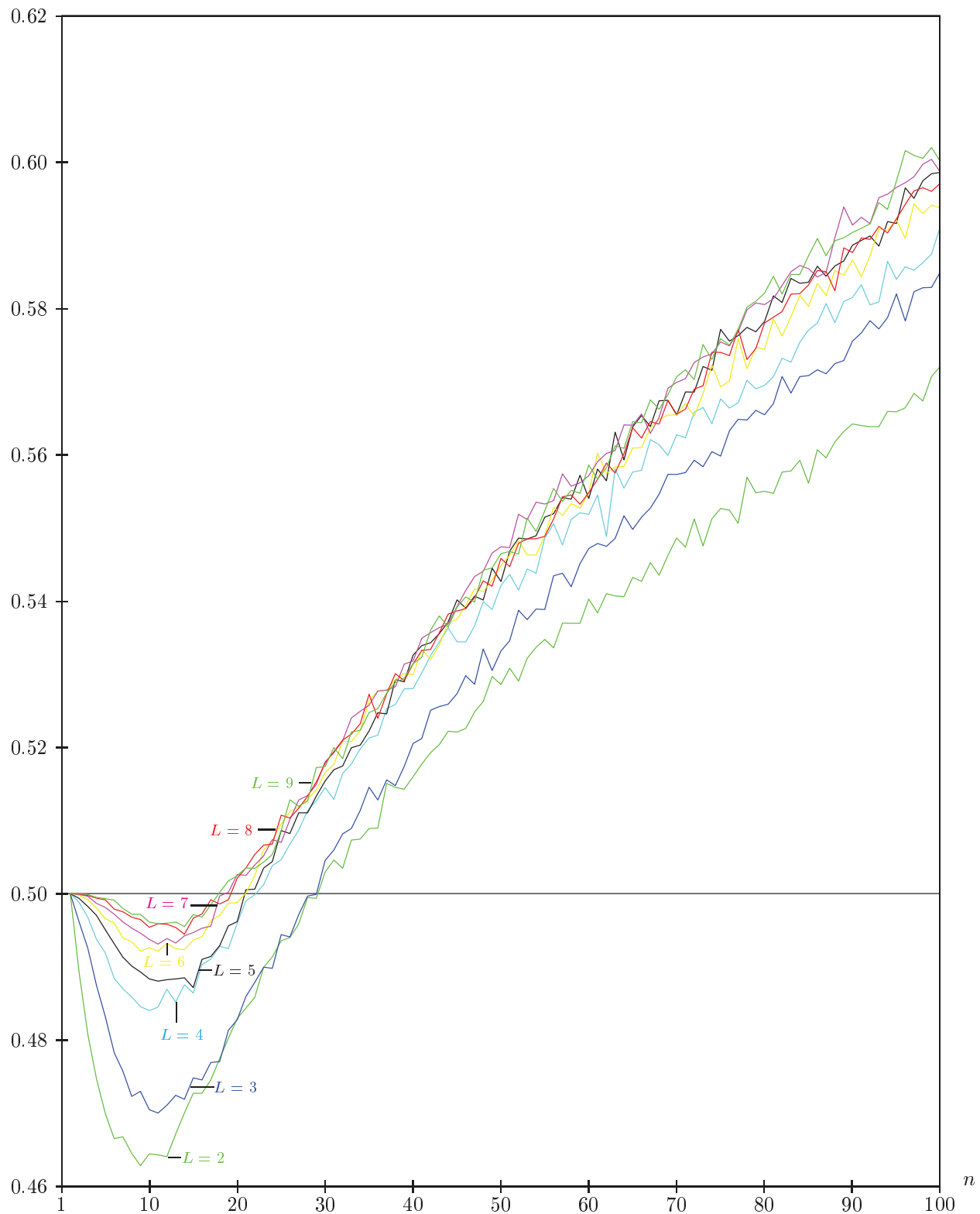


Abbildung 2.7: (**Max-3-SAT**) Erfolgskurven unter $r = (n, m^*)$, verschiedenen Gruppengrößen L und probabilistisch wiederholter zyklischer schneller lokaler Suche in Abhängigkeit von n

für $n = 100$, geringfügig unterschiedliche Werte der Erfolgsquoten auf. Diese liegen bei zyklischer schneller lokaler Suche fast sämtlich unter 0.60, bei voller lokaler Suche dagegen nur für $L \leq 4$. Die Erfolgsquoten scheinen also unter voller lokaler Suche etwas schneller anzusteigen. Vergleiche dazu auch die Erfolgsquoten unter größeren Problemparametern n und Gruppengröße 3 in Tabelle 2.3.

n	100	200	500	1000	2000
zyklisch schnell	0.584960	0.65080	0.80285	0.92950	0.98955
voll	0.589905	0.66425	0.81840	0.93450	0.99260

Tabelle 2.3: Erfolgsquoten unter Gruppengröße 3

Im Allgemeinen lassen höhere Erfolgsquoten jedoch nicht auf bessere Zielfunktionswerte der in den Mehrheitsläufen ermittelten lokal maximalen Lösungen schließen. Um dies einzusehen, werden die *durchschnittlich* erzielten f -Werte der lokal maximalen Lösungen aus allen Mehrheitsläufen voller lokaler Suche mit fester Gruppengröße herangezogen und mit den analogen Durchschnittswerten unter zyklischer schneller lokaler Suche verglichen. Tabelle 2.4 zeigt die Ergebnisse für $n = 100$, und zwar jeweils in einer Zeile für die wiederholte zyklische schnelle lokale Suche (ZS), die wiederholte volle lokale Suche (V) und die Differenz unter beiden Suchvarianten (–).

	zufällig	$L = 2$	$L = 3$	$L = 4$	$L = 5$	$L = 6$	$L = 7$	$L = 8$	$L = 9$
ZS	414.34	416.22	416.71	416.89	417.23	417.27	417.49	417.54	417.71
V	414.12	416.15	416.62	416.83	417.14	417.23	417.47	417.53	417.70
–	0.22	0.07	0.09	0.06	0.09	0.04	0.02	0.01	0.01

Tabelle 2.4: Durchschnittliche f -Werte gefundener lokal maximaler Lösungen

Man erkennt, dass die volle lokale Suche in allen Spalten geringfügig schlechter abschneidet, insbesondere in den zufälligen Läufen laut Spalte 1. Hier ist die Differenz am größten. Innerhalb der Teilfolgen der geraden bzw. ungeraden L fällt die Differenz monoton in L . Relativ zum Ausgangsmaterial aus den zufälligen Läufen gesehen, ist also der Nutzen der Mehrheitsbildung unter voller lokaler Suche größer. Diese Beobachtung erklärt das schnellere Ansteigen der Erfolgskurven unter der vollen lokalen Suche.

In Abbildung 2.8 sind für jedes $L \in \{2, 3, \dots, 9\}$ und $n \in \{1, 2, \dots, 100\}$ die Differenzen der durchschnittlichen f -Werte aus den Mehrheitsläufen unter Gruppengröße L und den mittleren f -Werten aus den zufälligen Läufen dargestellt. Die Graphik links zeigt die Daten der wiederholten vollen lokalen Suche, rechts der probabilistisch wiederholten zyklischen schnellen lokalen Suche. Die Daten stammen aus denselben Läufen, aus denen die Erfolgsquoten der Abbildungen 2.6 und 2.7 ermittelt worden sind. Aus optischen Gründen sind die Datenpunkte jeweils durch Linien verbunden.

Man erkennt, dass die Differenzen für jedes L und alle $n > 1$ positiv sind. Die durchschnittlichen f -Werte nach den Mehrheitsläufen sind also unter jeder Gruppengröße L besser als nach den zufälligen Läufen. Zudem zeigt sich eine Zunahme der Differenzwerte mit aufsteigender Gruppengröße. Bei genauerem Hinsehen kann man für jedes *gerade* L beobachten, dass der Qualitätssprung von L zu $L + 1$ sowohl größer als von $L - 1$ zu L als

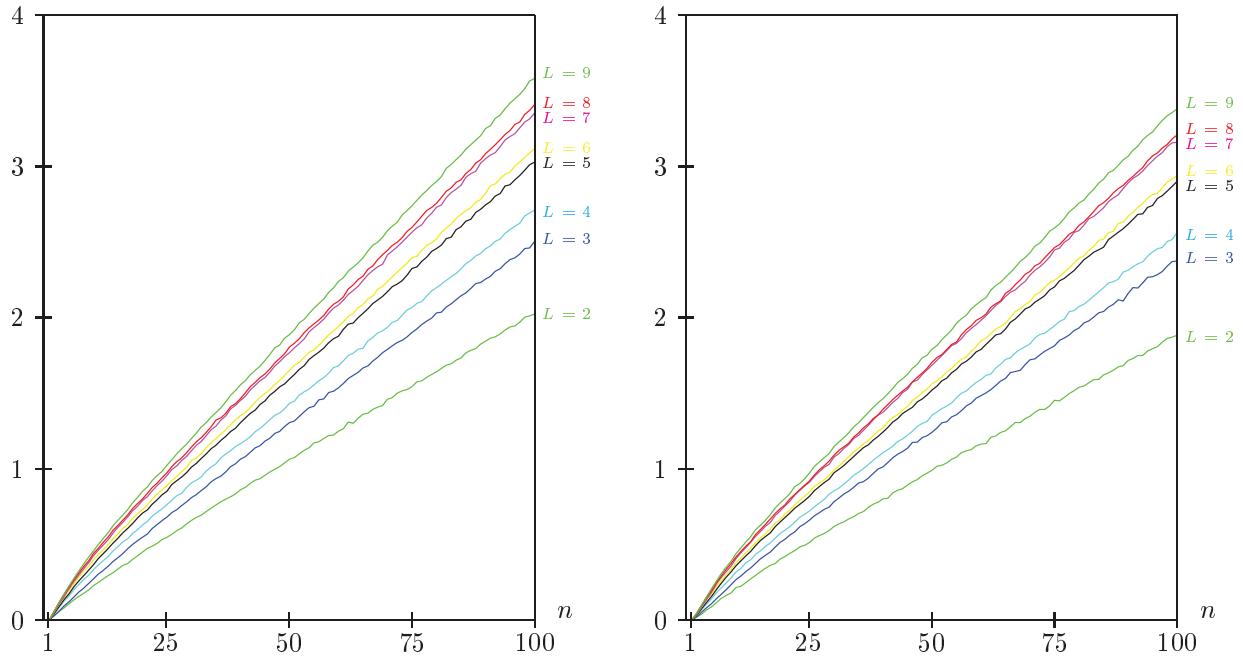


Abbildung 2.8: (**Max-3-SAT**) Differenzen der durchschnittlichen f -Werte lokal maximaler Lösungen aus Mehrheitsläufen verschiedener Gruppengrößen L und zufälligen Läufen bei zufällig gewählten Instanzen mit Parametern $r = (n, m^*(n))$ und wiederholter voller lokaler Suche (links) bzw. probabilistisch wiederholter zyklischer schneller lokaler Suche (rechts)

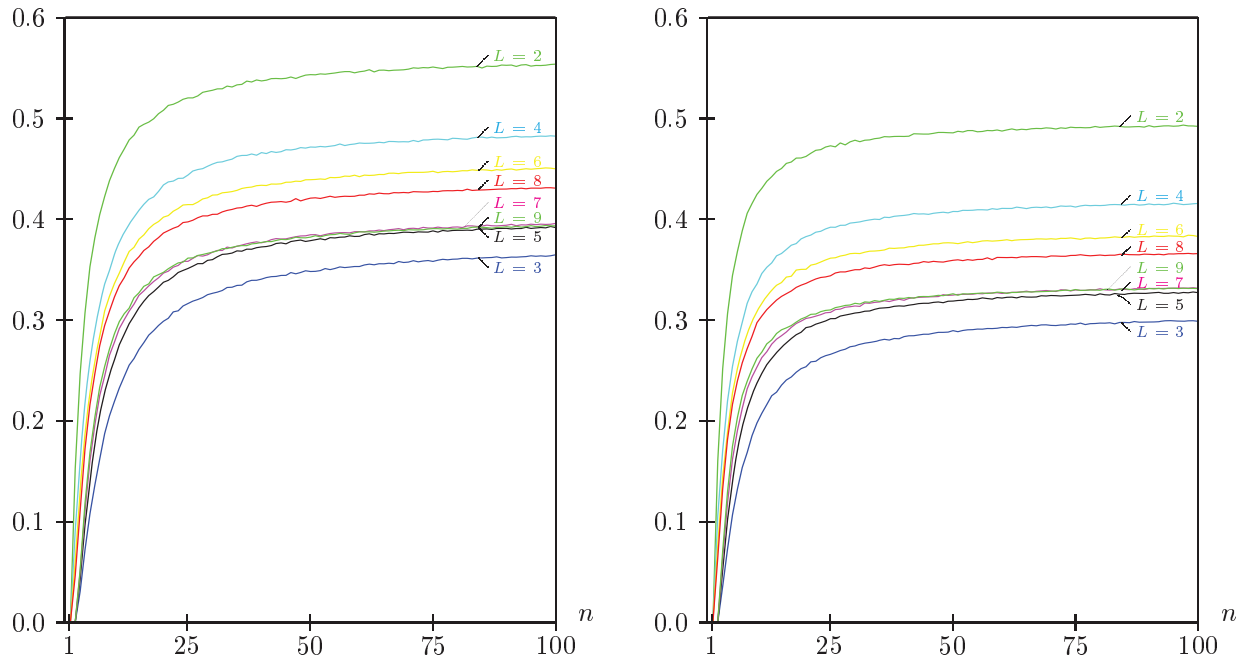


Abbildung 2.9: (**Max-3-SAT**) Kurven der Schrittzahlquotienten unter $r = (n, m^*)$ und wiederholter voller lokaler Suche (links) bzw. probabilistisch wiederholter zyklischer schneller lokaler Suche (rechts) in Abhängigkeit von n

auch von $L+1$ zu $L+2$ ist. Es tritt in gewisser Weise also eine Paarung der „Kurven“ auf. Man beachte dazu, dass die Mehrheitsvektoren bei ungeraden L eindeutig sind. Gerade

Gruppengrößen führen aufgrund des „Auswürfeln“ von unentschiedenen Komponenten zu Mehrheitsvektoren, welche teilweise nach denselben Regeln wie die zufällig gewählten Startvektoren der zufälligen Läufe gebildet sind. Dies erklärt das etwas schlechtere Abschneiden der geraden L , relativ zu den ungeraden L . Im Hinblick auf die Misserfolgstäler der Mehrheitsbildung ist bemerkenswert, dass die f -Wert-Differenzen *unter keinem* n negativ sind. Unter *jedem* Problemparameter $n > 1$ liefern Mehrheitsläufe im Durchschnitt bessere lokale Maxima als zufällige Läufe.

Abbildung 2.9 zeigt die Kurven der Schrittzahlquotienten für Max-3-SAT unter den Parametern n und $m^*(n)$ in Abhängigkeit von n . Die zugehörigen Zahlenwerte wurden aus denselben Läufen gewonnen, aus denen die Erfolgsquoten bzw. die durchschnittlichen f -Werte der gefundenen lokal maximalen Lösungen resultierten. Als erste Beobachtung sei festgehalten, dass alle Kurven mit Quotient 0 „starten“. Bei *beliebigen* Optimierungsaufgaben mit Lösungsraum $\{0, 1\}^1$ verschwinden die Schrittzahlen von Mehrheitsläufen unter jeder Gruppengröße L und mit Raum $\{0, 1\}^2$ unter jedem ungeraden L :

- a) Im Raum $\{0, 1\}^1$ sind nur zwei Situationen möglich: Entweder beide Lösungen sind lokal maximal; dann terminieren alle Läufe nach 0 Schritten im Startknoten. Oder es gibt nur eine lokal maximale Lösung; dann enden alle Suchläufe nach keinem oder einem Verbesserungsschritt trivialerweise in dieser Lösung, so dass der Mehrheitsvektor unter beliebigem L diese Lösung ist.
- b) Seien $x^\ell = (x_1^\ell, x_2^\ell) \in \{0, 1\}^2$ für $\ell = 1, \dots, L$ und $y = (y_1, y_2) \in \{0, 1\}^2$ der Mehrheitsvektor zu x^1, \dots, x^L , wobei L ungerade. Dann tritt y_i für $i = 1, 2$ in jeweils mindestens $\lceil \frac{L}{2} \rceil$ Komponenten von x_i^1, \dots, x_i^L auf. Nach dem Schubfachprinzip existiert folglich ein $\ell \in \{1, \dots, L\}$ mit $x_1^\ell = y_1$ und $x_2^\ell = y_2$, d.h. $x^\ell = y$.

Zudem lassen sich anhand Abbildung 2.9 folgende weitere Beobachtungen bzw. Vermutungen für Max-3-SAT aufstellen:

- (1) Alle Folgen von Schrittzahlquotienten sind monoton wachsend und konvergieren in \mathbb{R} mit Grenzwert deutlich kleiner 1. Man betrachte hierbei zusätzlich die Schrittzahlquotienten für größere n in Tabelle 2.5. Dass die Werte kleiner als 1 sind, ist ein Indiz dafür, dass die Mehrheitsvektoren typischerweise besser sind als zufällig gewählte Startvektoren. Bei der praktischen Durchführung der Verfahren konnten bei den Mehrheitsläufen auch deutlich kürzere Rechenzeiten beobachtet werden.

n	100	200	500	1000	2000
zyklisch schnell	0.298	0.304	0.307	0.308	0.308
voll	0.364	0.371	0.375	0.376	0.377

Tabelle 2.5: Schrittzahlquotienten unter Gruppengröße 3

- (2) Für jeweils fast alle n sind die Schrittzahlquotienten unter wiederholter voller lokaler Suche (V) und beliebigem L größer als die entsprechenden Schrittzahlquotienten unter wiederholter zyklischer schneller lokaler Suche (ZS), d.h.

$$\frac{S_{\text{Mh}}(n, m^*, L, \text{V})}{S_{\text{zf}}(n, m^*, \text{V})} > \frac{S_{\text{Mh}}(n, m^*, L, \text{ZS})}{S_{\text{zf}}(n, m^*, \text{ZS})}$$

bzw.

$$\frac{S_{\text{Mh}}(n, m^*, L, \text{ZS})}{S_{\text{Mh}}(n, m^*, L, \text{V})} < \frac{S_{\text{zf}}(n, m^*, \text{ZS})}{S_{\text{zf}}(n, m^*, \text{V})}.$$

Diese Beobachtung steht im Einklang damit, dass die Differenz der durchschnittlichen f -Werte aus zyklischer schneller und voller Suche nach den Mehrheitsläufen für beliebiges L geringer ist als nach den zufälligen Läufen.

- (3) Bei fest gewählter Variante lokaler Suche gilt: An jeweils fast allen Stellen n nehmen die Schrittzahlquotienten unter geraden L größere Werte als unter ungeraden L an. Dies ist insofern naheliegend, als bei *geraden* Gruppengrößen in den Mehrheitsvektoren im Allgemeinen zufällig gewählte Komponenten auftreten. Solche Startvektoren werden also in einem gewissen Anteil ihrer Komponenten wie die Startvektoren der zufälligen Läufe gebildet. Die zugehörigen Schrittzahlquotienten liegen dementsprechend näher bei 1. Innerhalb der geraden Gruppengrößen L sinkt mit wachsendem L die Wahrscheinlichkeit, dass einzelne Komponenten zufällig gewählt werden. Die Schrittzahlquotienten werden bei festem n und wachsendem geraden L folglich kleiner. Ferner lässt sich bei wachsendem *ungeraden* L eine Zunahme der Schrittzahlquotienten beobachten.³

Für tiefer liegende, sichere Beobachtungen müssten an dieser Stelle umfangreichere und womöglich genauere Daten mittels höherer Instanzenzahlen B , insbesondere für größere L , erhoben werden. Eine intensivere Beurteilung der Schrittzahlen gehört aber nicht zu den Hauptanliegen dieser Arbeit.

Es wird nun untersucht, wie die Mehrheitsbildung statt unter der in Abhängigkeit von n fest gewählten Klauselzahl $m^*(n)$ unter *beliebiger* Klauselzahl m abschneidet. Unter denselben Varianten lokaler Suche wie oben und fest gewählter Gruppengröße 3 wurden dazu zunächst die Erfolgsquotenfolgen für alle $(n, m) \in \{10, 20, 30, \dots, 100\} \times \{1, 2, 3, \dots, 500\}$ ermittelt. Für jedes solche Paar (n, m) wurden 100 000 Instanzen unabhängig und zufällig erzeugt. Die Erfolgskurven in Abhängigkeit von m sind für jede dieser Variablenzahlen n in den Abbildungen 2.10 und 2.11 dargestellt. Die Scharen der Erfolgskurven unterscheiden sich zwischen wiederholter voller und probabilistisch wiederholter zyklischer schneller lokaler Suche nur äußerst geringfügig. Bei genauem Hinsehen erkennt man etwa am rechten Rand der Abbildungen, also bei $m = 500$, dass die Erfolgsquoten unter voller lokaler Suche etwas höhere Werte annehmen. Diese Beobachtung ist bereits im Fall $m = m^*$ aufgetreten und dort diskutiert worden.

Es ist wenig überraschend, dass sich die Erfolgskurven in Abhängigkeit von m für festes $n \in \{10, 20\}$ unter beiden Varianten lokaler Suche vollständig unterhalb 0.50 bewegen. Man erinnere sich, dass die Stellen $n = 10$ und $n = 20$ jeweils im Misserfolgstal der Mehrheitsbildung unter den Parametern $L = 3$ und $m = m^*$ liegen.⁴ Unter $n = 30, 40, \dots, 100$ überschreiten die Folgen dagegen die 0.5-Hürde für nicht zu große m . Auch das passt zu den Beobachtungen unter $m = m^*$. Dass auch die Folgen unter $n = 40, 50, \dots, 100$ für größere m wieder fallen, ist vermutlich darin begründet, dass die Anzahl lokal maximaler

³zumindest bis $L = 7$. Bei wiederholter voller lokaler Suche traten allerdings unter $L = 9$ ab einem gewissen n überwiegend kleinere Schrittzahlquotienten als unter $L = 7$ auf.

⁴vgl. blau skizzierte Erfolgskurven in den Abbildungen auf den Seiten 41 und 42

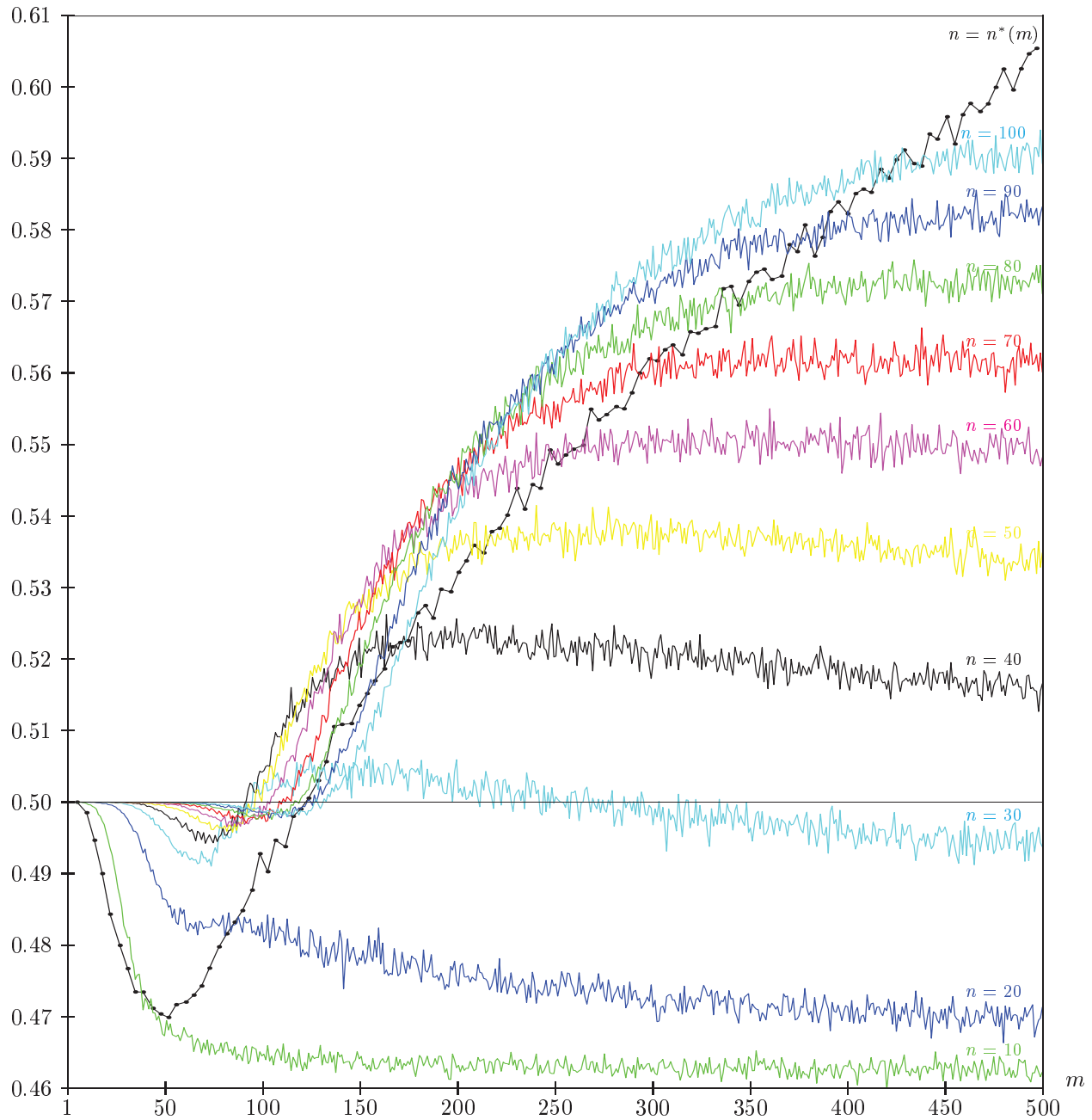


Abbildung 2.10: **(Max-3-SAT)** Erfolgskurven unter $r = (n, m)$, $L = 3$ und wiederholter voller lokaler Suche in Abhängigkeit der Klauselzahl m

Lösungen bei festem n mit wachsendem m abnimmt. Sämtliche Suchläufe terminieren dann unabhängig von der Art des Startknotens in diesen wenigen Stellen, so dass sich die Erfolgsquoten dem Wert 0.5 annähern. Ferner sind für $n = 30, 40, \dots, 100$ Misserfolgstäler zu erkennen, die mit wachsendem n flacher und länger werden.

Die wohl interessanteste Beobachtung ist, dass die Erfolgskurven für $n = 30, 40, \dots, 100$ in der Nähe von $m = m^*(n)$ jeweils einen maximalen Wert annehmen. Um dies einzusehen, sind in beiden Abbildungen zusätzlich die Erfolgsquoten unter $(n, m^*(n))$ mit schwarzen Knoten und Verbindungslinien skizziert. Zu vorgegebenem m bezeichnet $n^*(m)$ dasjenige

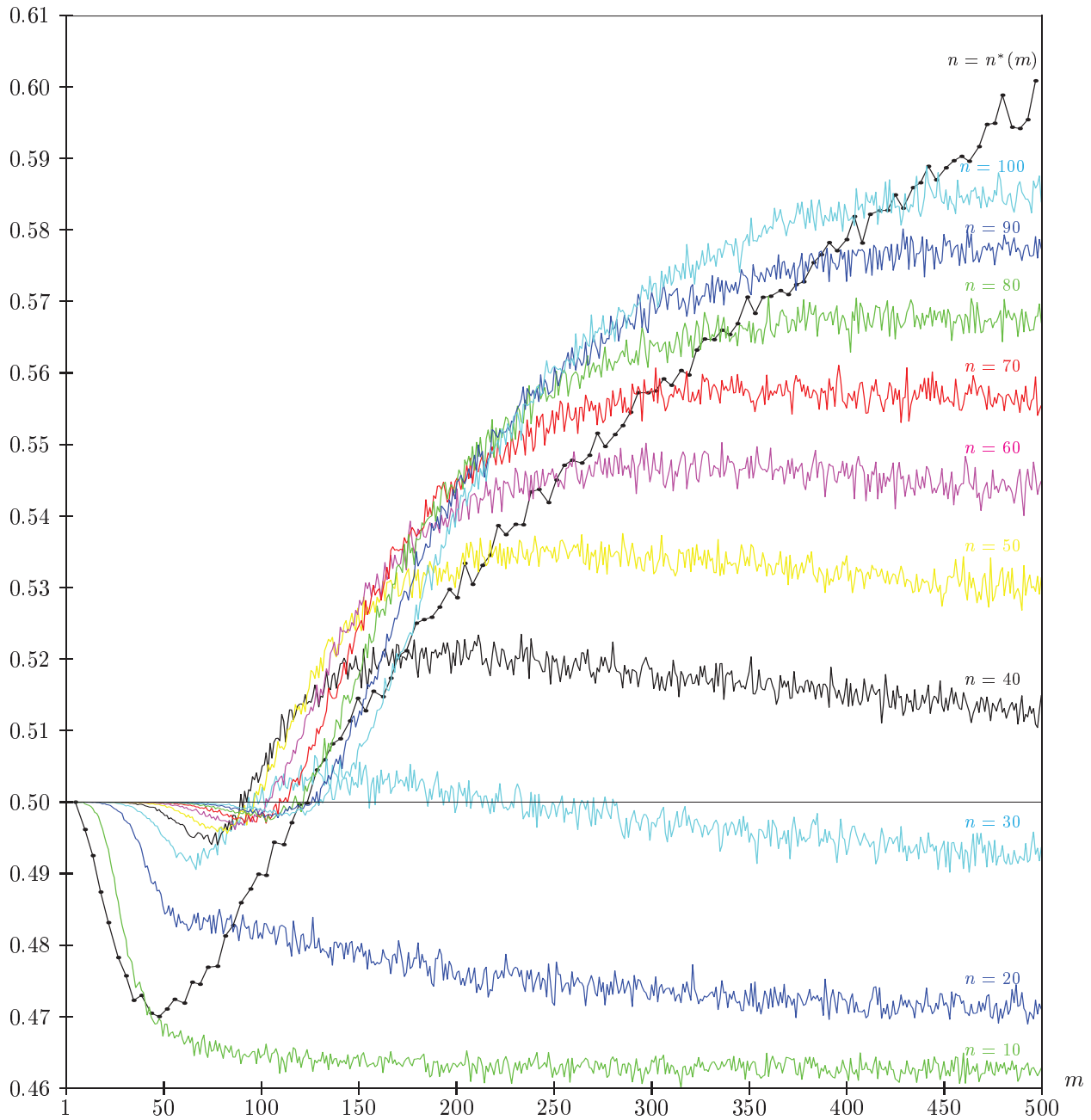


Abbildung 2.11: **(Max-3-SAT)** Erfolgskurven unter $r = (n, m)$, $L = 3$ und probabilistisch wiederholter zyklischer schneller lokaler Suche in Abhängigkeit der Klauselzahl m

n , für das $m^*(n) = m$ gilt, d.h.

$$n^*(m) = \lfloor (4.25)^{-1} m \rfloor.$$

Je größer n , umso mehr scheinen die Maximalstellen der „bunten Kurven“ von $m^*(n)$ abzuweichen, desto stärker ist also der Begriff „in der Nähe“ zu relativieren. Nun lassen sich zwei Gegenargumente einwenden:

1. Die Klauselzahlen $m(n)$ aus [27] (vgl. S. 39) verhalten sich nur asymptotisch wie $4.25n$, sind aber tatsächlich etwas größer als $4.25n$. So gilt etwa $m(20) = 91$ und

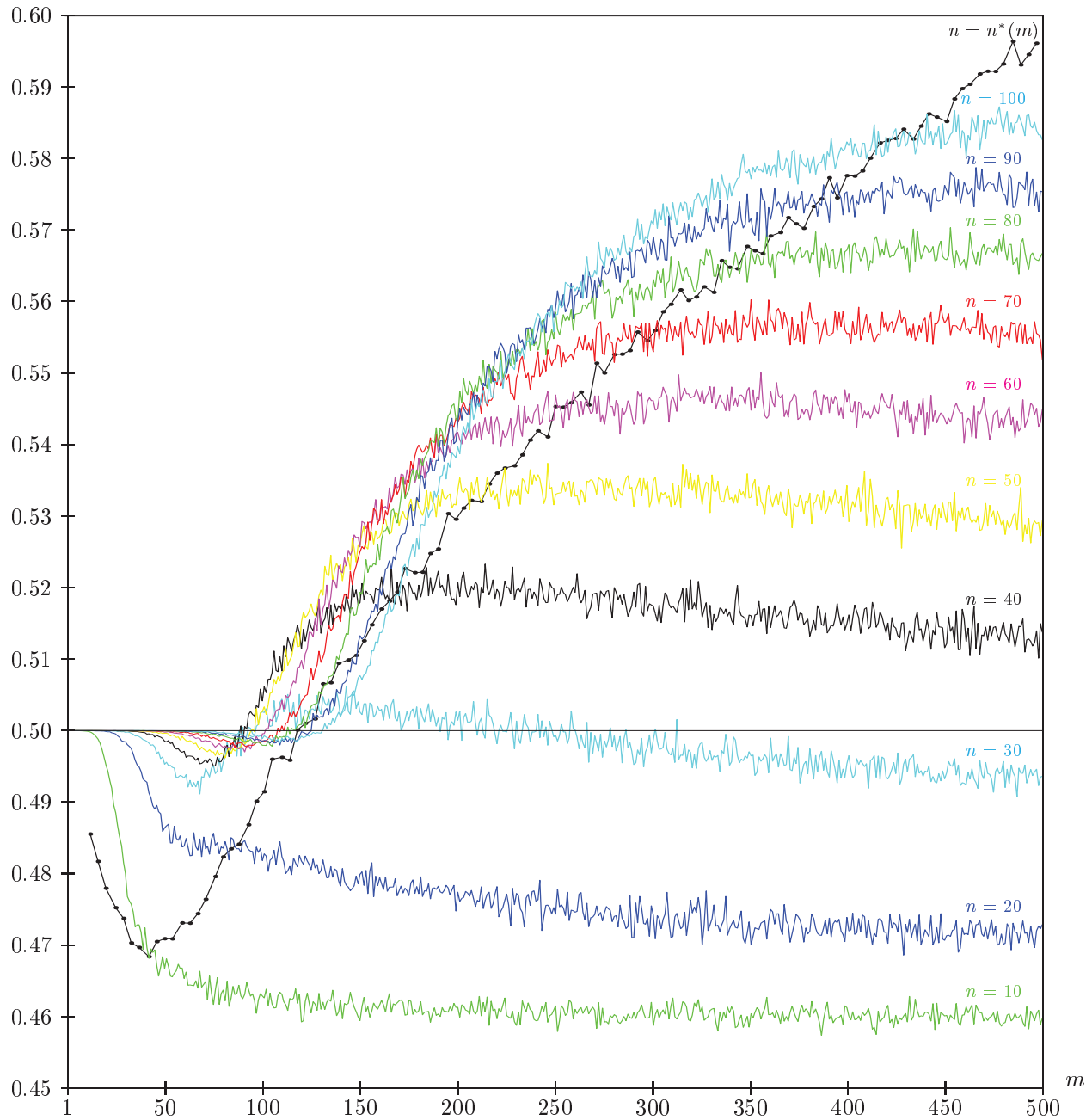


Abbildung 2.12: **(Max-3-SAT)** Erfolgskurven unter $r = (n, m)$, $L = 3$ und probabilistisch wiederholter zyklischer schneller lokaler Suche in Abhängigkeit der Klauselzahl m für das Modell „reiner Klauseln“

$m(50) = 215$. Betrachtet man die Erfolgsquoten jedoch an den Stellen $m(n)$, so erkennt man, dass auch diese nur ungefähr den Maximalstellen entsprechen.

2. In [27] werden nur Instanzen betrachtet, deren Klauseln in folgendem Sinne „rein“ sind: Keine der $n \geq k$ Variablen darf in einer Klausel mehrfach auftreten. Unter probabilistisch wiederholter zyklischer schneller lokaler Suche wurden für das Max-3-SAT-Modell „reiner Klauseln“ die gleichen Berechnungen wie für obiges Modell „gemischter Klauseln“ durchgeführt. Dabei wurden zufällige Instanzen wie folgt gebildet: Für jede

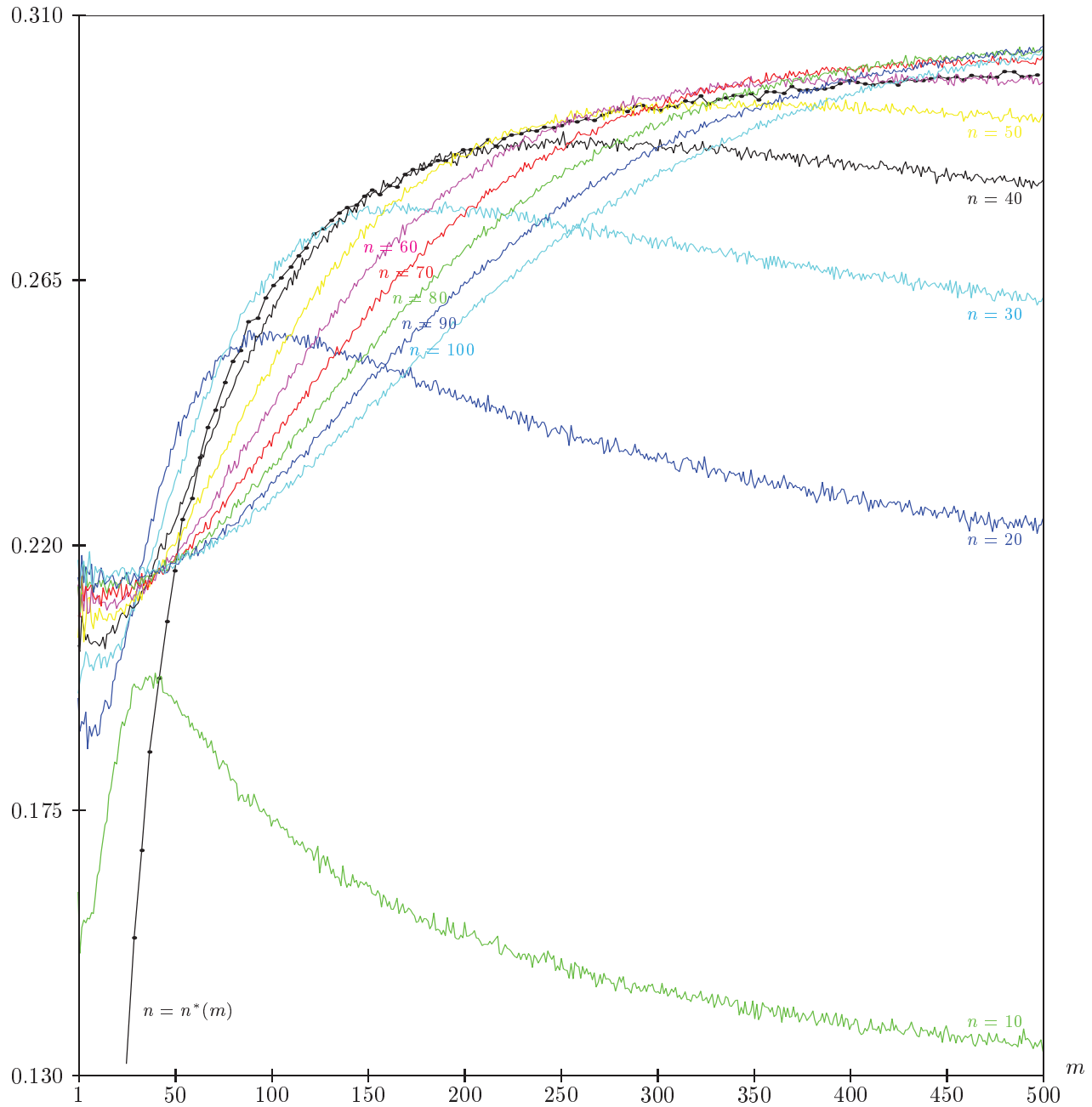


Abbildung 2.13: **(Max-3-SAT)** Kurven der Schrittzahlquotienten unter $r = (n, m)$, $L = 3$ und probabilistisch wiederholter zyklischer schneller lokaler Suche in Abhängigkeit der Klauselzahl m für das Modell „gemischter Klauseln“

der m Klauseln wurden zunächst die n Variablen zufällig gemäß der Gleichverteilung permutiert. Gemäß dieser Permutation wurden dann die ersten drei Variablen übernommen und jede Variable mit Wahrscheinlichkeit $\frac{1}{2}$ negiert. Abbildung 2.12 zeigt die ermittelten Erfolgskurven für dieses Modell. Ein Vergleich mit Abbildung 2.11 macht deutlich, dass die Wahl des Klauselmodells keinen wesentlichen Einfluss auf die relative Lage der Maximalstellen zu den Werten $m^*(n)$ ausübt.

Die Maximalstellen der Folgen der Erfolgsquoten in m stimmen also nur grob mit den

Werten $m(n)$ überein. „Modulo dieser Ungenauigkeit“ soll an dieser Stelle trotzdem als Beobachtung festgehalten werden, dass die Mehrheitsbildung besonders erfolgreich ist, wenn die Problem instanzen in gewissem Sinne schwierig sind.

Interessant sind für beide Modelle auch die Kurven der Schrittzahlquotienten in Abhängigkeit von m . Die Kurven unter $(n, m) \in \{n^*, 10, 20, 30, \dots, 100\} \times \{1, 2, 3, \dots, 500\}$, probabilistisch wiederholter zyklischer schneller lokaler Suche und Gruppengröße 3 sind in Abbildung 2.13 dargestellt. Da sich die entsprechenden Kurven unter wiederholter voller lokaler Suche in ihrem qualitativen Verlauf nur unwesentlich von diesen unterscheiden, soll hier eine einzige Darstellung genügen. Auch bei Beschränkung auf „reine“ Klauseln ergibt sich praktisch dasselbe Bild der Kurven. Man erkennt, dass jede Folge unter festem n nach oben beschränkt ist, ihren maximalen Wert an genau einer Stelle m annimmt und für größere Klauselzahlen wieder fällt. Für $n = 10$ liegt diese Stelle bei $m = m^*(10) = 43$, für $n = 20, 30, \dots, 100$ jeweils etwas über $m^*(n)$, wobei die Abweichung von $m^*(n)$ mit n wächst. Trotzdem ist wohl die Beobachtung gerechtfertigt, dass für jede Zahl n an Variablen maximal viele Schritte in den Mehrheitsläufen, relativ zu den zufälligen Läufen, durchgeführt werden, wenn die Klauselzahl m in der Nähe von $m^*(n)$ liegt. „Schwierige“ Problemparameter bewirken also hohe Schrittzahlquotienten und gute Aussichten auf Erfolg der Mehrheitsbildung. Man beachte, dass hohe Schrittzahlquotienten lediglich für große Schrittzahlen von Mehrheitsläufen *relativ* zu zufälligen Läufen stehen. Im Allgemeinen gehen hohe Schrittzahlquotienten jedoch *nicht* mit großen absoluten Schrittzahlen der Mehrheitsläufe einher. Dies demonstrieren etwa die Folgen $(S_{zf}(n, m, ZS))_{m=1,2,\dots}$ und $(S_{Mh}(n, m, 3, ZS))_{m=1,2,\dots}$ für jeweils festes $n \in \{10, 20, 30, \dots, 100\}$ in Abbildung 2.14. Außer für $n = 10$ bei den Mehrheitsläufen sind alle Kurven im gesamten skizzierten Bereich monoton wachsend.

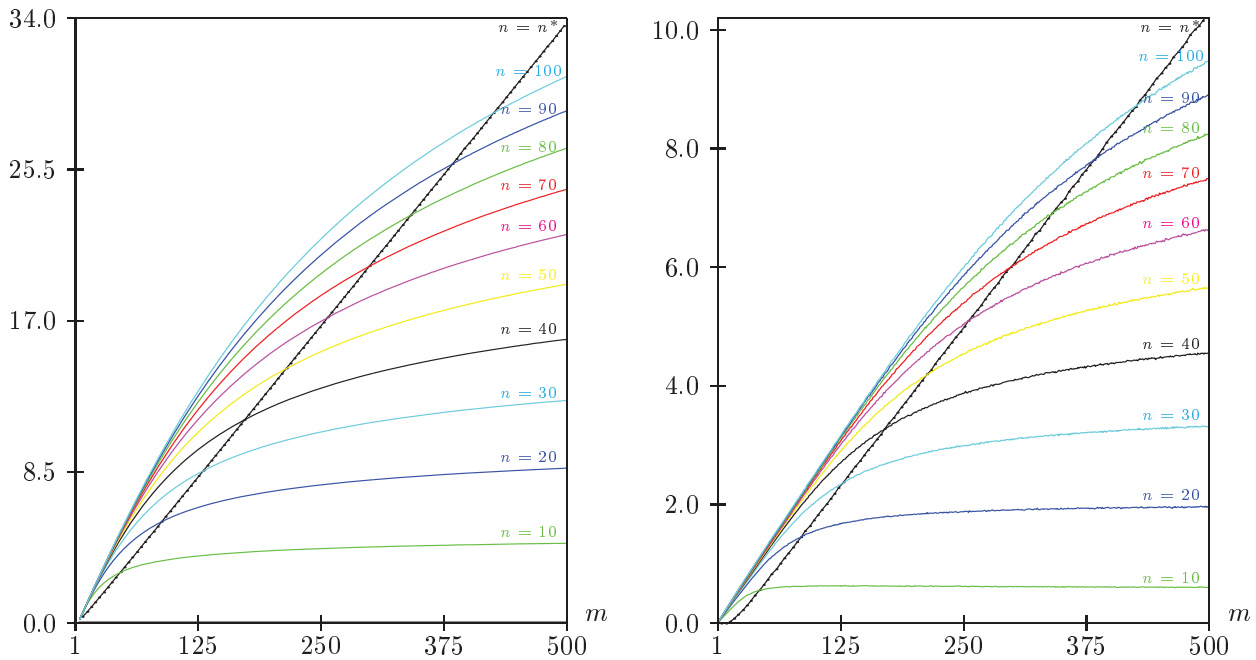


Abbildung 2.14: (**Max-3-SAT**) Folgen der mittleren Schrittzahlen unter $r = (n, m)$, $L = 3$ und probabilistisch wiederholter zyklischer schneller lokaler Suche in Abhängigkeit der Klauselzahl m für zufällige Läufe (links) und Mehrheitsläufe (rechts)

2.2.2 Erfolgsquoten für das Dynamische Optimierungsproblem

Es seien $n \geq 2$ und $C(t) = (c_{ij}(t))_{i,j=0,1}$ für $t = 1, \dots, n-1$ Matrizen mit Komponenten in $[0, 1[$. n ist der Problemparameter, $C(1), C(2), \dots, C(n-1)$ bilden eine Probleminstanz aus $I(n)$. Eine solche Instanz wird zufällig gewählt, indem jede der insgesamt $4 \cdot (n-1)$ Matrixeinträge aus $[0, 1[$ zufällig gleichverteilt gewählt wird.⁵ Gesucht ist eine Lösung $x = (x_1, \dots, x_n) \in \{0, 1\}^n$ mit möglichst großem f -Wert $f(x) = \sum_{t=1}^{n-1} c_{x_t x_{t+1}}(t)$.

Hauptziel dieses Abschnittes ist die Untersuchung der Erfolgskurven in Abhängigkeit von n für verschiedene Gruppengrößen L . Zur praktischen Ermittlung lokal maximaler Lösungen wurde ausschließlich die probabilistisch wiederholte zyklische schnelle lokale Suche verwendet. Abbildung 2.15 zeigt die Erfolgskurven unter $L = 2, 3, \dots, 9$. Für $n \in \{2, 3, 4, \dots, 500\}$ wurden je 100 000 Instanzen zufällig und unabhängig gewählt. Die Berechnung der Erfolgsquoten unter $L = 3$ basierte für jedes $n \in \{501, 502, \dots, 1000\}$ auf 100 000 Instanzen und für $n \in \{1010, 1020, 1030, \dots, 5000\} \cup \{5100, 5200, 5300, \dots, 50000\}$ auf 10 000 Instanzen.

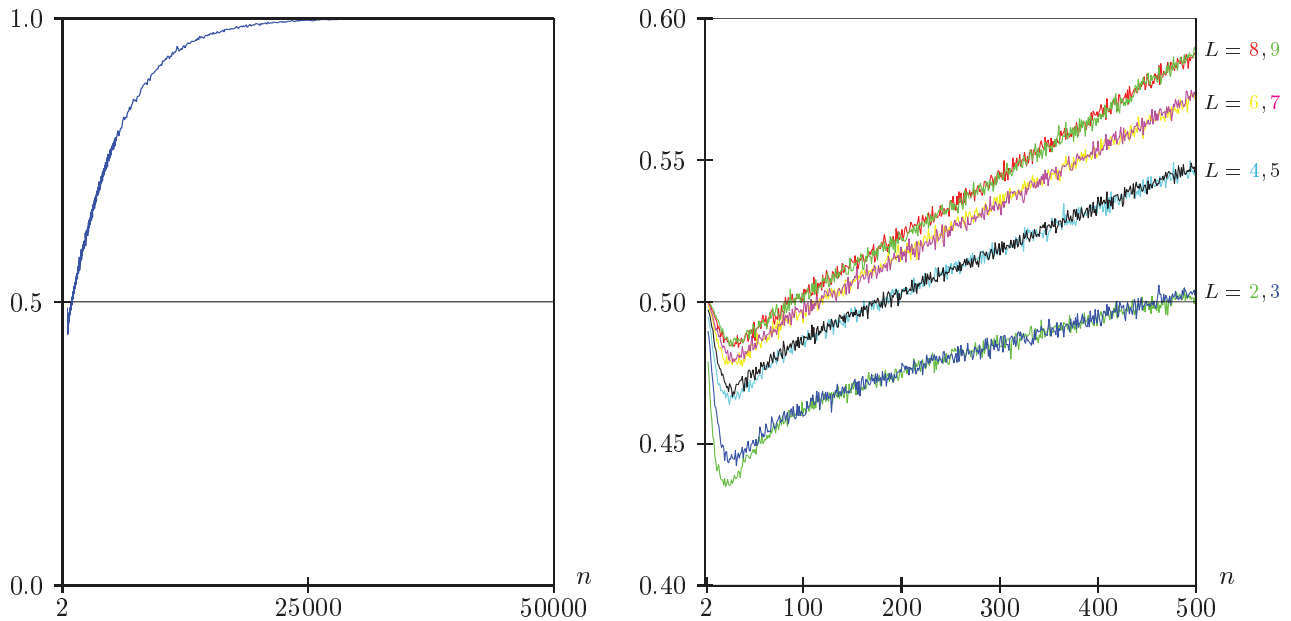


Abbildung 2.15: **(Dynamisches Optimierungsproblem)** Erfolgskurven in Abhängigkeit von n unter $L = 3$ (links) bzw. $L = 2, 3, \dots, 9$ (rechts) und probabilistisch wiederholter zyklischer schneller lokaler Suche

Es zeigt sich, dass die Erfolgsquoten unter L und $L + 1$ (L gerade) für schließlich alle n nahezu dieselben Werte ergeben. Je zwei Erfolgskurven zu $L, L + 1$ fallen also ab einer bestimmten Stelle zusammen. Unter festem n wachsen die Erfolgsquoten monoton in L (L gerade).

Obwohl sich die Ermittlung der Erfolgsquoten auf eine einzige Variante wiederholter lokaler Suche beschränkt, wird folgende verallgemeinerte Vermutung geäußert:

⁵Wegen der Endlichkeit des darstellbaren Zahlenbereichs aus $[0, 1[$ ist die praktisch behandelbare Instanzenmenge $I(n)$ zum Parameter n endlich.

VERMUTUNG 2.5

Der Erfolg von Mehrheitsbildung für das Dynamische Optimierungsproblem in Abhängigkeit von n konvergiert unter jeder Gruppengröße $L \geq 2$ und beliebiger in Abschnitt 1.1 genannter Variante wiederholter lokaler Suche für $n \rightarrow \infty$ gegen 1.

Wie schon bei Max-3-SAT treten auch hier Misserfolgstäler auf: Unter jedem festen L sind die Erfolgsquoten in einem Intervall kleiner Zahlen n kleiner als 0.50. Die durchschnittlichen f -Werte in den gefundenen lokal maximalen Lösungen sind dagegen nach den Mehrheitsläufen unter jedem $n > 2$ besser als nach den zufälligen Läufen: Abbildung 2.16 links zeigt die Differenzen der mittleren erzielten f -Werte aus den Mehrheitsläufen und den zufälligen Läufen. Man erkennt, dass die Differenzen, abgesehen vom linken Rand bei $n = 2$, alle positiv sind. Wie schon bei Max-3-SAT nehmen die Differenzen bei festgehaltenem Problemparameter n von geradem L zu ungeradem $L + 1$ schwächer zu als von ungeradem L zu geradem $L + 1$ (vgl. Abb. 2.8, S. 44).

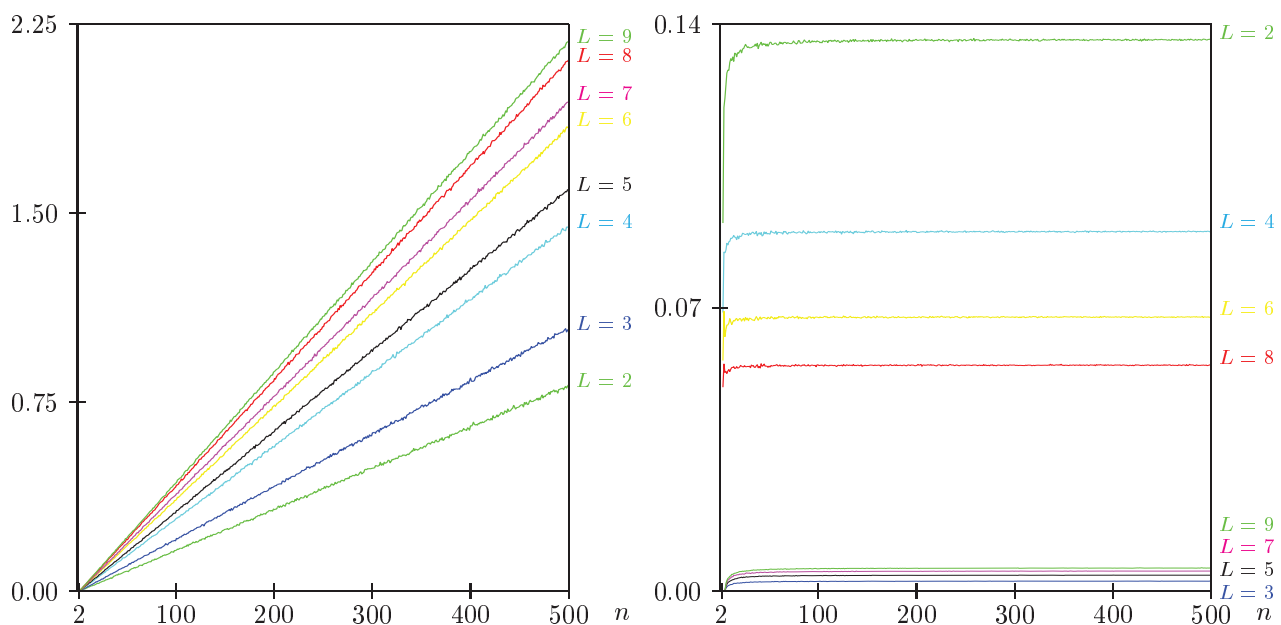


Abbildung 2.16: **(Dynamisches Optimierungsproblem)** Differenzen der durchschnittlichen f -Werte lokal maximaler Lösungen aus Mehrheitsläufen verschiedener Gruppengrößen L und zufälligen Läufen (links) und Kurven der Schrittzahlquotienten in Abhängigkeit von n (rechts)

Schließlich sind in Abbildung 2.16 rechts die Kurven der Schrittzahlquotienten in Abhängigkeit von n unter den Gruppengrößen 2, 3, ..., 9 dargestellt. Hier lassen sich analoge Beobachtungen wie bei Max-3-SAT machen (vgl. Abb. 2.9, S. 44). Insbesondere sind alle Quotienten wesentlich kleiner als 1. Mehrheitsläufe machen also im Mittel deutlich weniger Schritte als zufällige Läufe.

2.2.3 Erfolgsquoten für das Shiftregister-Problem

Gegeben sei eine natürliche Zahl $n > 3$, welche als Problemparameter fungiert. Zu n lassen sich Probleminstanzen $I = \{v_1, v_2, v_3\} \subseteq \{1, \dots, n-1\}$ mit $\#I = 3$ wie folgt zu-

fällig bestimmen: Man permutiere die Zahlen $1, \dots, n-1$ zufällig gemäß der Gleichverteilung auf S_{n-1} und übernehme dann die entsprechend dieser Anordnung ersten drei Werte v_1, v_2, v_3 . Gesucht ist eine Lösung $x = (x_0, \dots, x_{n-1}) \in \{0, 1\}^n$ mit minimalem Defekt $f(x)$. $f(x)$ ist die Anzahl an Komponenten x_i , $i \in \{0, 1, \dots, n-1\}$, welche nicht der Antimehrheit aus den durch die Vorgängerpositionen v_1, v_2, v_3 festgelegten Komponenten $x_{(i-v_1) \bmod n}$, $x_{(i-v_2) \bmod n}$, $x_{(i-v_3) \bmod n}$ von x entsprechen. Zur Ermittlung lokal minimaler Lösungen wurde für diesen Abschnitt stets die probabilistisch wiederholte zyklische schnelle lokale Suche verwendet. Zudem beschränkten sich die durchgeführten Experimente auf Mehrheitsbildung unter Gruppengröße 3.

Die spezielle Struktur des Shiftregister-Problems liefert einen Äquivalenzbegriff auf dem Lösungsraum $V = \{0, 1\}^n$: Für $x = (x_0, \dots, x_{n-1}) \in V$ und $d \in \{0, 1, \dots, n-1\}$ bezeichne

$$x \gg^d := (x_{(0+d) \bmod n}, x_{(1+d) \bmod n}, \dots, x_{(n-1+d) \bmod n}) \in V$$

die durch **Drehen** um d aus x hervorgegangene Lösung. Dann ist durch

$$x \sim y \iff \text{es gibt ein } d \in \{0, \dots, n-1\} \text{ mit } x \gg^d = y \text{ oder } 1 - x \gg^d = y$$

für $x, y \in V$ offensichtlich eine Äquivalenzrelation \sim über V definiert. $x, y \in V$ sind genau dann äquivalent, wenn sie durch Drehen um d Positionen für ein geeignetes d oder durch Drehen um d und anschließendem **Flippen**, d.h. Vertauschen von 0 und 1 in jeder Komponente, auseinander hervorgehen. Äquivalente Lösungen haben denselben Defekt.

Um die Äquivalenzen zu berücksichtigen, kann man die Mehrheitsbildung wie folgt variieren: Seien $L \geq 2$ Lösungen $x^1, \dots, x^L \in V$ gegeben. Dann sei die zufällige Wahl eines Mehrheitsvektors x^{Mh} zu x^1, \dots, x^L , wie die Mehrheitsbildung bis dato zu verstehen war (vgl. Definition 1.1), mit (V0) bezeichnet. Alternativ dazu bietet sich an, statt x^1, \dots, x^L jeweils solche Repräsentanten der Äquivalenzklassen $[x^1], \dots, [x^L]$ zur Bildung von x^{Mh} heranzuziehen, die „möglichst viel Struktur gemeinsam“ haben. Um diese Idee zu konkretisieren, sei $d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ definiert durch

$$d(x, y) := \sum_{i=0}^{n-1} |x_i - y_i|$$

für alle $x = (x_0, \dots, x_{n-1}), y = (y_0, \dots, y_{n-1}) \in \mathbb{R}^n$. Falls speziell $x, y \in V$, ist $d(x, y)$ der HAMMING-Abstand von x und y , d.h. die Zahl der Komponenten, in denen sich x und y unterscheiden (vgl. S. 16). Für die Bestimmung von x^{Mh} sind folgende pragmatische Varianten denkbar:

- (V1) Setze $y^1 := x^1$. Für $i = 2, 3, \dots, L$ wähle $y^i \in [x^i]$ so, dass $d(y^i, x^1)$ minimal ist. Wähle für x^{Mh} einen Mehrheitsvektor zu y^1, \dots, y^L gemäß (V0).

Statt x^1 kann natürlich jede beliebige Lösung aus $\{x^1, \dots, x^L\}$ als Referenzlösung benutzt werden. Da es sich bei x^1, \dots, x^L im Kontext der wiederholten lokalen Suche jedoch um lokal minimale Lösungen aus L unabhängigen Läufen handelt, ist die Wahl der Referenzlösung ohne Bedeutung. Führt man das Verfahren allerdings für *jede* Lösung als Referenzlösung durch, kann man sich anschließend etwa für diejenige Referenzlösung entscheiden,

die zu einem minimalen durchschnittlichen HAMMING-Abstand von y^1, \dots, y^L führt. Diese Variante ist mit nicht vertretbarem Zeitaufwand verbunden und wurde daher nicht implementiert. In einer weiteren Variante (V2) werden die zur Mehrheitsbildung herangezogenen Lösungen nicht wie in (V1) nach einer festen Referenzlösung, sondern die i -te Lösung nach dem Schwerpunkt der bis dahin festgelegten $i - 1$ Lösungen „ausgerichtet“:

(V2) Setze $z^1 := x^1$. Für $i = 2, 3, \dots, L$ wähle $z^i \in [x^i]$ so, dass $d(z^i, \frac{1}{i-1}(z^1 + \dots + z^{i-1}))$ minimal ist. Wähle für x^{Mh} einen Mehrheitsvektor zu z^1, \dots, z^L gemäß (V0).

Statt in der Reihenfolge $1, 2, \dots, L$ kann in (V2) gemäß jeder beliebigen Permutation von $\{1, \dots, L\}$ vorgegangen werden. Das bringt aber, wiederum aufgrund der Unabhängigkeit der L Läufe wiederholter lokaler Suche, keine Vorteile.

Die Hoffnung besteht nun darin, dass die Erfolgskurven von Mehrheitsbildung in Abhängigkeit von n unter allen drei Varianten (V0), (V1) und (V2) für den Einsatz von Mehrheitsbildung sprechen. Darüber hinaus sollten die Varianten (V1) und (V2) den Erfolg gegenüber (V0) noch steigern. Die Erfolgskurven im Bereich $4 \leq n \leq 5000$ bzw. $4 \leq n \leq 100$ sind in Abbildung 2.17 skizziert. Unter Variante (V0) wurden die Erfolgsquoten für $n \in \{4, 5, \dots, 100\} \cup \{105, 110, \dots, 300\} \cup \{310, 320, \dots, 3000\} \cup \{3050, 3100, \dots, 5000\}$ sowie unter (V1), (V2) für $n \in \{4, 5, \dots, 100\} \cup \{105, 110, \dots, 300\} \cup \{350, 400, \dots, 3000\} \cup \{3100, 3200, \dots, 5000\}$ berechnet. Dafür wurden jeweils 100 000 Probleminstanzen unabhängig und zufällig gewählt, falls $n \leq 100$, bzw. 10 000 Instanzen, falls $n > 100$.

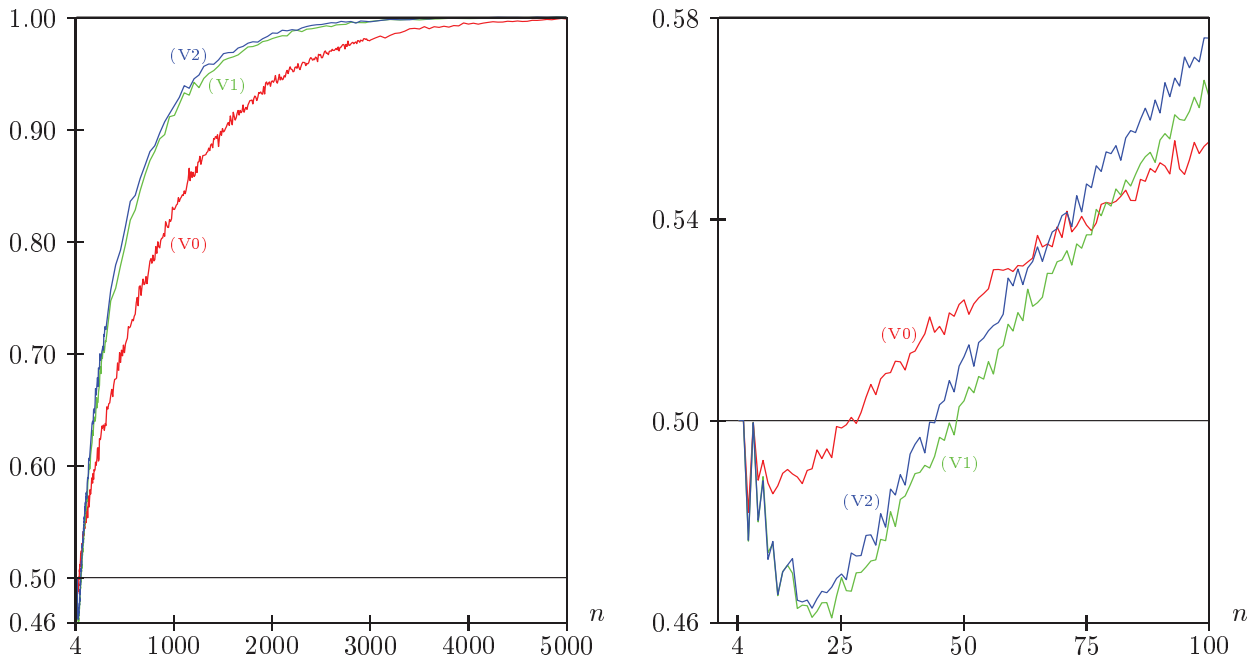


Abbildung 2.17: **(Shiftregister-Problem)** Erfolgskurven in Abhängigkeit von n unter Gruppengröße 3 und probabilistisch wiederholter zyklischer schneller lokaler Suche für Mehrheitsbildung verschiedener Varianten

Die Erfolgskurven sprechen wie bei Max-3-SAT und dem Dynamischen Optimierungsproblem allesamt eindeutig für den Einsatz von Mehrheitsbildung für schließlich alle n :

VERMUTUNG 2.6

Der Erfolg von Mehrheitsbildung für das Shiftregister-Problem in Abhängigkeit von n konvergiert unter jeder Gruppengröße $L \geq 2$, jeder der Varianten (V0), (V1), (V2) von Mehrheitsbildung und beliebiger in Abschnitt 1.1 genannter Variante wiederholter lokaler Suche für $n \rightarrow \infty$ gegen 1.

Unter allen drei Varianten ist für kleine n ein Misserfolgstal zu erkennen, welches unter (V1) und (V2) länger und tiefer ist als unter (V0). Eine mögliche Erklärung hierfür wird in Abschnitt 2.3.1 gegeben. Ab einer bestimmten Zahl n_0 sind die Erfolgsquoten unter festem $n \geq n_0$ unter Variante (V1) deutlich höher als unter (V0) sowie unter (V2) noch geringfügig höher als unter (V1). Die Erfolgskurve unter (V2) konvergiert also schneller als unter (V1) und unter (V1) schneller als unter (V0) gegen 1.

Betrachte nun die durchschnittlichen Defekte der lokal minimalen Lösungen aus den Läufen lokaler Suche, getrennt nach zufälligen Läufen und Mehrheitsläufen (Abbildung 2.18). Für $n = 4$ existiert genau eine Instanz, nämlich $\{v_1, v_2, v_3\} = \{1, 2, 3\}$. Hier gilt $\min f = 0$, und es existieren genau 6 lokal minimale Lösungen, nämlich die beiden Elemente der Äquivalenzklasse $[(0, 1, 0, 1)]$ und die vier Elemente von $[(0, 0, 1, 1)]$, alle mit Defekt 0. Unter dieser höchst einfachen Problem Instanz terminiert folglich jeder Lauf lokaler Suche mit global minimalem f -Wert. Insbesondere verschwindet auch der Durchschnittswert. Für $n = 5$ gibt es vier Problem Instanzen. Unter jeder dieser Instanzen existiert genau eine Äquivalenzklasse lokal minimaler Lösungen, jeweils mit Defekt 1. Der mittlere f -Wert lokal minimaler Lösungen beliebiger Läufe ist hier also 1. Zudem sind natürlich die Startlösungen der Mehrheitsläufe unter den Varianten (V1) und (V2) bei Vorliegen einer einzigen Äquivalenzklasse lokal minimaler Lösungen a priori lokal minimal. Folglich verschwinden die Schrittzahlen der Mehrheitsläufe unter $n = 5$ und (V1) bzw. (V2). Ab $n = 6$ existieren auch Instanzen, deren lokal minimale Lösungen nicht sämtlich global minimal sind. Die durchschnittlichen Defekte können sich also für $n \geq 6$ prinzipiell von Lauftyp zu Lauftyp unterscheiden.

Alle „Kurven“ zeigen einen regelmäßigen Zickzack-Verlauf. Ist $n \geq 6$ gerade, so sind die durchschnittlichen f -Werte innerhalb eines festen Lauftyps an den jeweils ungeraden Stellen $n + 1$ größer als an Stelle n und auch größer als an der jeweils ungeraden Stelle $n - 1$. Falls zudem $(n \bmod 12) \neq 6$, scheinen die mittleren f -Werte auch an den jeweils ungeraden Stellen $n - 1$ größer als an Stelle n zu sein. Möglicherweise bildet für die letzte Beobachtung auch die Stelle $n = 10$ eine Ausnahme. Hier waren die mittleren f -Werte unter (V2) rund 1.6173, bei $n = 9$ jedoch rund 1.6098, also etwas kleiner.

Die entscheidende Beobachtung an dieser Stelle ist, dass sich für fast alle n die mittleren Defekte von zufälligen Läufen über Mehrheitsläufe nach (V0) und (V1) hin zu (V2) jeweils verbessern. Dies ist insbesondere für den größten Teil der Parameter n des Misserfolgstals der Fall. Für kein einziges n sind die Durchschnittswerte der Mehrheitsläufe größer als die der zufälligen Läufe.

Schließlich sind in den beiden Graphiken in Abbildung 2.19 die Kurven der Schrittzahlquotienten in Abhängigkeit von n dargestellt, und zwar jeweils für alle drei Varianten (V0), (V1) und (V2). Die linke Graphik zeigt die Kurven im Bereich $4 \leq n \leq 5000$. In der rechten

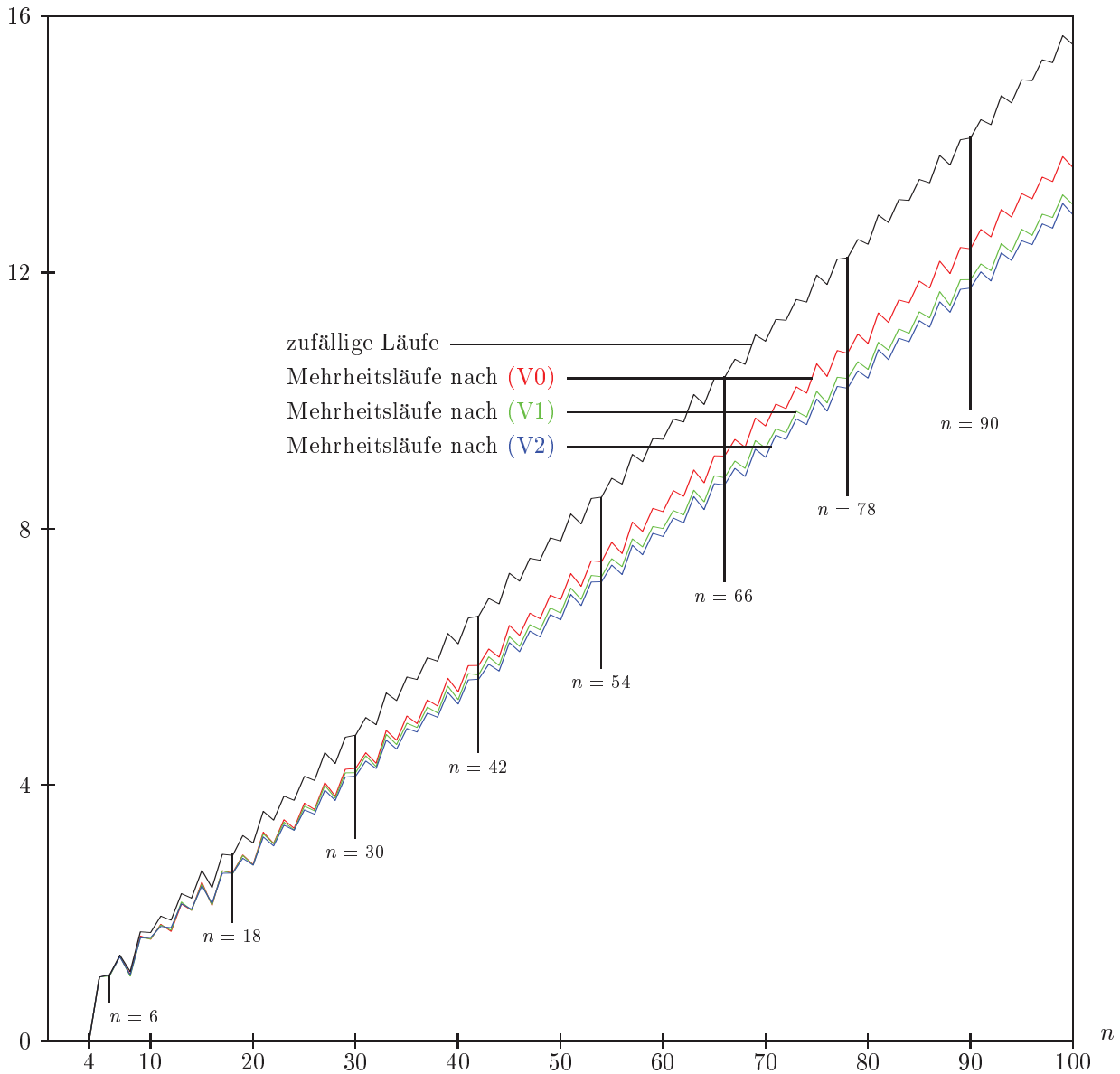


Abbildung 2.18: **(Shiftregister-Problem)** Durchschnittliche Defekte lokal minimaler Lösungen bei probabilistisch wiederholter zyklischer schneller lokaler Suche und verschiedenen Varianten der Mehrheitsbildung

Graphik ist der Ausschnitt für $4 \leq n \leq 100$ vergrößert wiedergegeben. Die Schrittzahlquotienten unter (V1) und (V2) sind offenbar für alle n kleiner als unter (V0). Zudem sind sie unter (V2) für alle hinreichend großen n kleiner als unter (V1). Ferner lässt sich beobachten, dass alle drei Kurven beschränkt sind und in \mathbb{R} gegen einen Wert deutlich kleiner 1 konvergieren. Bei der Durchführung der Experimente waren bei den Mehrheitsläufen auch deutlich kürzere Rechenzeiten als bei den zufälligen Läufen zu beobachten. Mehrheitsvektoren, gleich welcher Variante, als Startlösungen für Läufe lokaler Suche führen also nicht nur zu durchschnittlich besseren f -Werten, sondern auch zu kürzeren Laufzeiten. Man beachte allerdings, dass unter den Varianten (V1) und (V2) der größte Zeitanteil in die Bildung des Mehrheitsvektors investiert wird.

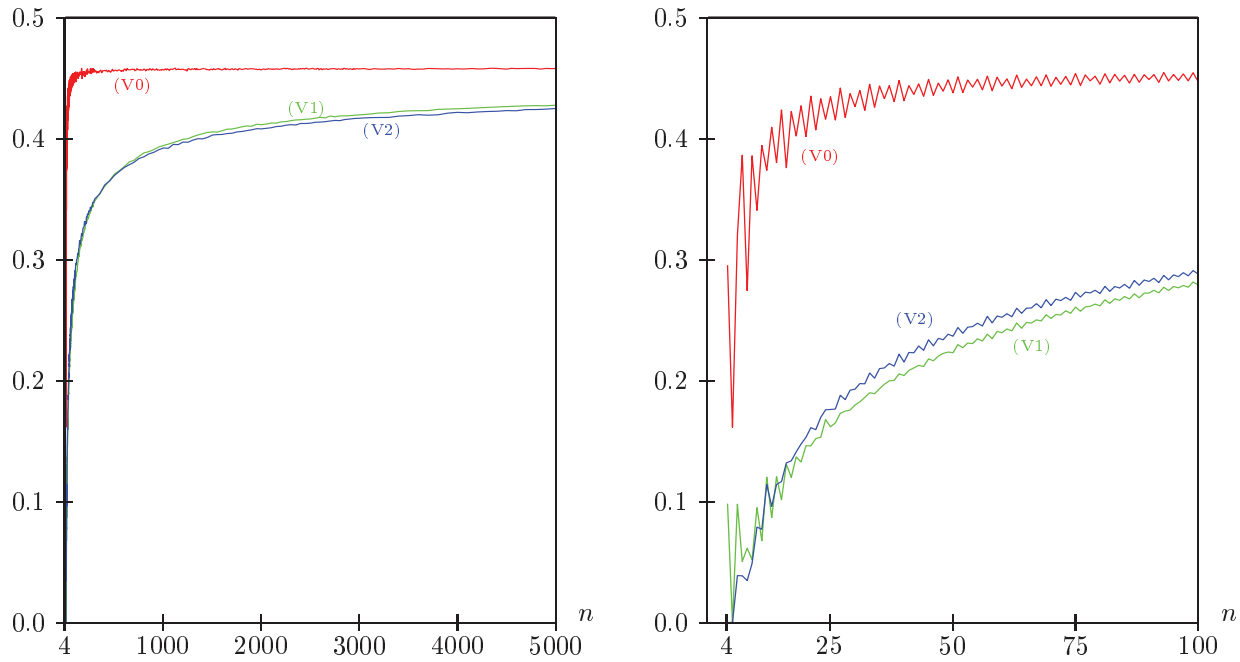


Abbildung 2.19: **(Shiftregister-Problem)** Kurven der Schrittzahlquotienten in Abhängigkeit von n unter probabilistisch wiederholter zyklischer schneller lokaler Suche und verschiedenen Varianten der Mehrheitsbildung

2.2.4 Erfolgsquoten für das Ising-Spinglas-Problem

Instanzen für das ± 1 -Ising-Spanglas-Problem auf dem $(n \times n)$ -Torus ohne äußeres Feld lassen sich auf einfache Weise zufällig erzeugen: Man bestimmt zu jeder der $2n^2$ Toruskanten $\{u, v\}$ eine Interaktion $J_{uv} \in \{-1, +1\}$, indem jeder der beiden Werte mit Wahrscheinlichkeit $\frac{1}{2}$ gewählt wird. n wird dabei als Problemparameter aufgefasst. Die Optimierungsaufgabe besteht darin, einen Spinzustand $x = (S_v)_v$ der n^2 Torusknoten zu finden, so dass

$$f(x) = - \sum_{\{u,v\}} J_{uv} S_u S_v$$

möglichst klein ist. Es wurden die probabilistisch sowie die deterministisch wiederholte zyklische schnelle lokale Suche verwendet. Die deterministische Variante ist bezüglich der in Abschnitt 1.2.4, S. 27, angegebenen Bijektion φ vom Lösungsraum in die Menge $\{0, 1\}^{n^2}$ zu verstehen. Die n^2 HAMMING-Nachbarn eines aktuellen Spinzustands x wurden also zeilenweise und innerhalb einer Zeile spaltenweise der Reihe nach untersucht. Zu jedem Parameter $n \in \{3, 4, \dots, 50\}$ wurden 100 000 Instanzen unabhängig und zufällig erzeugt. In den Abbildungen 2.20 bzw. 2.21 sind die Erfolgskurven bzw. die Kurven der Schrittzahlquotienten in Abhängigkeit von n unter den Gruppengrößen $L = 2, 3, \dots, 9$ dargestellt. Ferner wurden die Berechnungen speziell unter Gruppengröße 3 im Bereich $n \in \{55, 60, \dots, 200\}$ fortgesetzt. Abbildung 2.22 zeigt die entsprechenden Resultate.

Schließlich wird durch

$$x \sim y \iff y = x \text{ oder } y = -x$$

für je zwei Spinzustände x, y offensichtlich eine Äquivalenzrelation \sim über dem Lösungs-

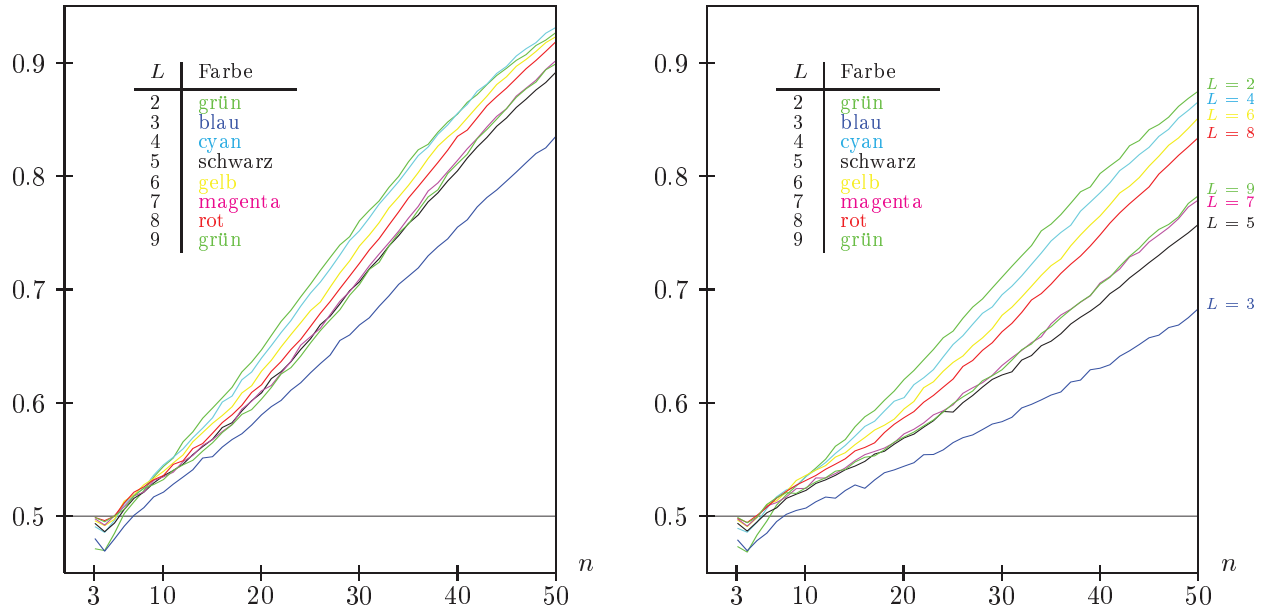


Abbildung 2.20: **(Ising-Spinglas-Problem)** Erfolgskurven in Abhängigkeit von n unter Gruppengrößen $L = 2, 3, \dots, 9$ und probabilistisch (links) bzw. deterministisch (rechts) wiederholter zyklischer schneller lokaler Suche

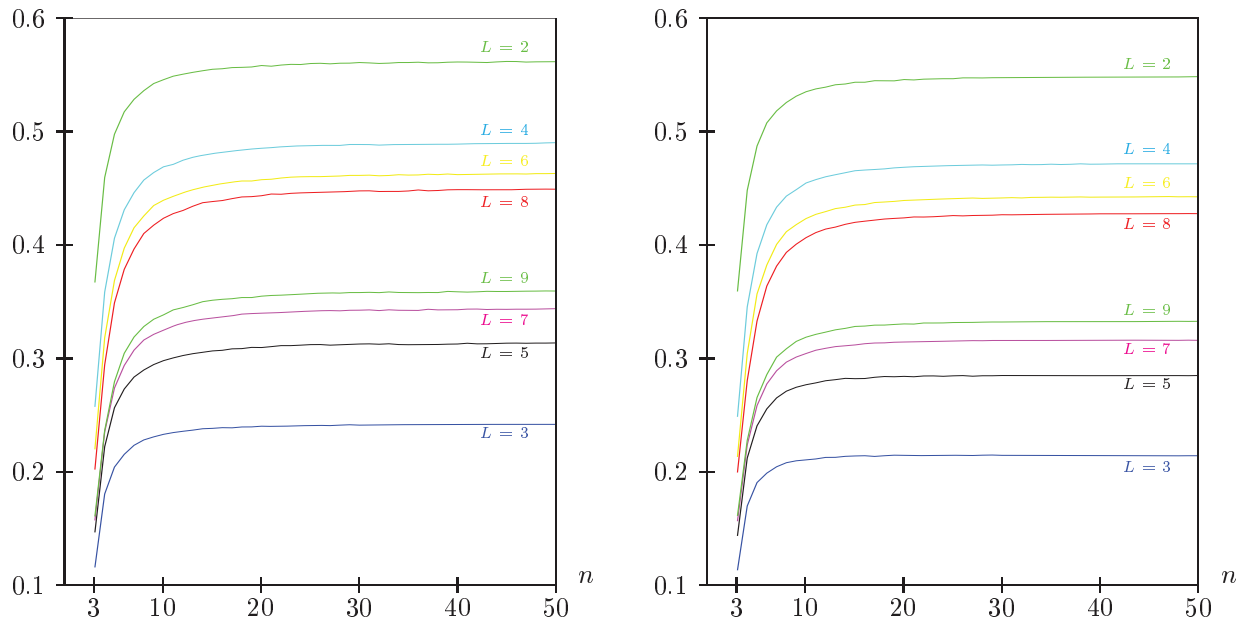


Abbildung 2.21: **(Ising-Spinglas-Problem)** Kurven der Schrittzahlquotienten in Abhängigkeit von n unter Gruppengrößen $L = 2, 3, \dots, 9$ und probabilistisch (links) bzw. deterministisch (rechts) wiederholter zyklischer schneller lokaler Suche

raum festgelegt. Zwei äquivalente Spinzustände sind also identisch oder gehen durch **Flippen**, d.h. Vertauschen von -1 und $+1$ in jedem Spin, auseinander hervor. Bezüglich dieses Äquivalenzbegriffs wurden analoge Varianten (V0), (V1), (V2) von Mehrheitsbildung wie

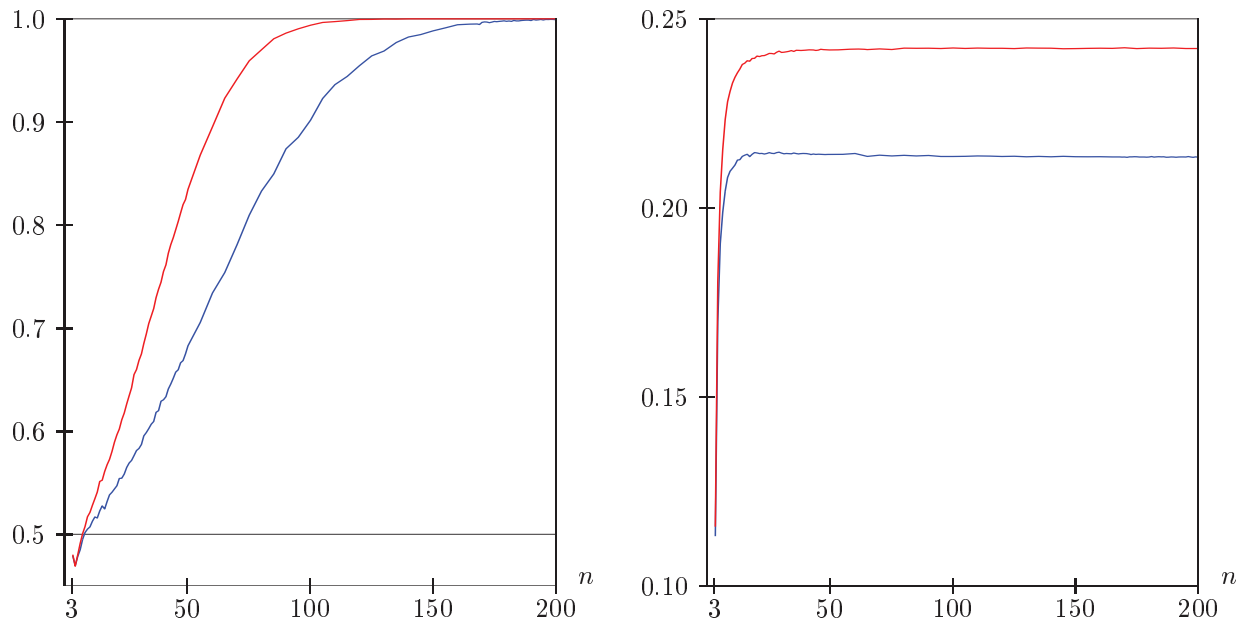


Abbildung 2.22: **(Ising-Spinglas-Problem)** Erfolgskurven (links) und Kurven der Schrittzahlquotienten (rechts) in Abhängigkeit von n unter Gruppengröße 3 und probabilistisch (blau) bzw. deterministisch (rot) wiederholter zyklischer schneller lokaler Suche

beim Shiftregister-Problem in Abschnitt 2.2.3 untersucht. Eine vergleichende Darstellung der Ergebnisse unter Gruppengröße 3 findet sich in Abbildung 2.23.

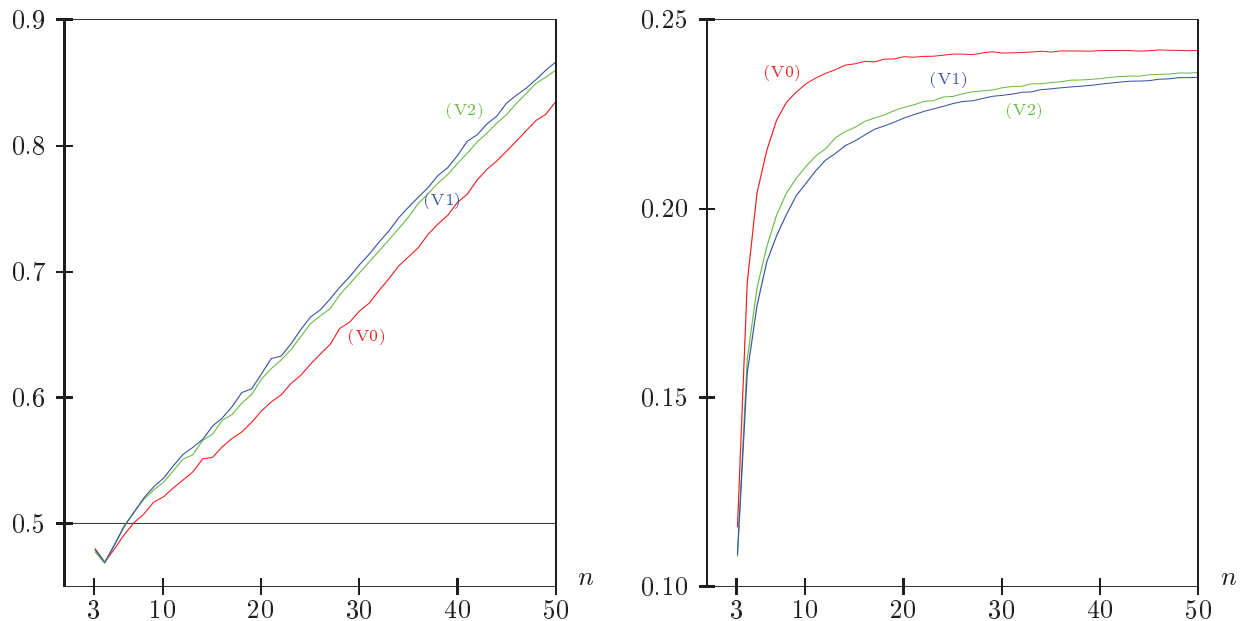


Abbildung 2.23: **(Ising-Spinglas-Problem)** Erfolgskurven (links) und Kurven der Schrittzahlquotienten (rechts) in Abhängigkeit von n unter Gruppengröße 3 und probabilistisch wiederholter zyklischer schneller lokaler Suche für Mehrheitsbildung verschiedener Varianten

Auf die Darstellung der durchschnittlichen Funktionswerte lokal minimaler Spinzustände soll an dieser Stelle aus Platzgründen verzichtet werden. Analog zu den Optimierungsproblemen der vorangegangenen Abschnitte sind die Durchschnittswerte nach den Mehrheitsläufen besser als nach den zufälligen Läufen, und zwar unter jeder Gruppengröße L und jedem Problemparameter n . Dies gilt insbesondere für die kleinen Parameter der Misserfolgstäler.

Alle für das Ising-Spینگlas-Problem berechneten Schrittzahlquotienten sind deutlich kleiner als 1. Zudem geben die Beobachtungen Anlass zu folgender

VERMUTUNG 2.7

Der Erfolg von Mehrheitsbildung für das Ising-Spینگlas-Problem in Abhängigkeit von n konvergiert unter jeder Gruppengröße $L \geq 2$, jeder der Varianten (V0), (V1) und (V2) von Mehrheitsbildung und beliebiger in Abschnitt 1.1 genannter Variante wiederholter lokaler Suche für $n \rightarrow \infty$ gegen 1.

2.2.5 Erfolgsquoten für das Rucksack-Problem

Zur Ermittlung empirischer Erfolgsquoten für das Rucksack-Problem wurden folgende Größen verwendet: Die Zahl n der Pakete sei der Problemparameter. Weiter sei M die größte, mittels der verwendeten Rechnerarchitektur darstellbare ganze Zahl. Hier wurde $M := 2^{63} - 1$ gesetzt. W sei die größte natürliche Zahl, die für die Größen s_1, \dots, s_n bzw. Werte v_1, \dots, v_n der Pakete erlaubt ist. Es sei $W := \lfloor \sqrt{Mn^{-1}} \rfloor$. Die zufällige Wahl einer Probleminstanz mit Parameter n ist dann als unabhängige und in der Menge $\{1, 2, \dots, W\}$ gleichverteilte Wahl von s_i und v_i , $i = 1, \dots, n$, zu verstehen. Zudem wird als Rucksackgröße $S := \lfloor \frac{n}{4} \rfloor \cdot W$ gewählt. S ist also etwa die Hälfte der erwarteten Summe aller Paketgrößen. Schließlich sei $C := W + 1$. Wegen $C > W \geq \max\{v_i \mid i = 1, \dots, n\}$ ist C als Konstante des Strafterms zulässig (vgl. Abschnitt 1.2.5). Sei weiter $x = (x_1, \dots, x_n) \in \{0, 1\}^n$. Falls $s(x) \leq S$, gilt $f(x) = v(x) \leq n \cdot W \leq n \cdot Mn^{-1} = M$. Ist dagegen $s(x) > S$, erhält man

$$\begin{aligned} f(x) &= v(x) - C(s(x) - S) < v(x) \leq M, \\ f(x) &= v(x) - C(s(x) - S) > -C s(x) > -n C^2 \geq -n M n^{-1} = -M. \end{aligned}$$

In jedem Fall ist $f(x)$ Element des darstellbaren Zahlenbereichs $\{-2^{63}, \dots, 2^{63} - 1\}$.

Um lokal maximale Lösungen zu erhalten, wurde ausschließlich die probabilistisch wiederholte zyklische schnelle lokale Suche eingesetzt. Für $n > 1$ wechselten sich die Läufe lokaler Suche unter der HAMMING-Nachbarschaft mit Läufen unter der in Abschnitt 1.2.5 beschriebenen Doppel-Flip-Nachbarschaft ab. Bei letzterer Nachbarschaft sind in Abschnitt 1.1.3, Algorithmus 1.5, die n HAMMING-Nachbarn $y^1(x), y^2(x), \dots, y^n(x)$ einer Lösung $x \in \{0, 1\}^n$ durch die $\binom{n}{2}$ Nachbarn $y^{ij}(x)$ mit $1 \leq i < j \leq n$ zu ersetzen (vgl. S. 29). Eine kanonische Reihenfolge dieser Nachbarn ergibt sich durch lexikographische Anordnung der Index-Paare (i, j) .

In Abbildung 2.24 sind die empirischen Erfolgsquoten und Schrittzahlquotienten von Mehrheitsbildung unter n und Gruppengröße 3 dargestellt. Für $n = 1, 2, \dots, 200$ und

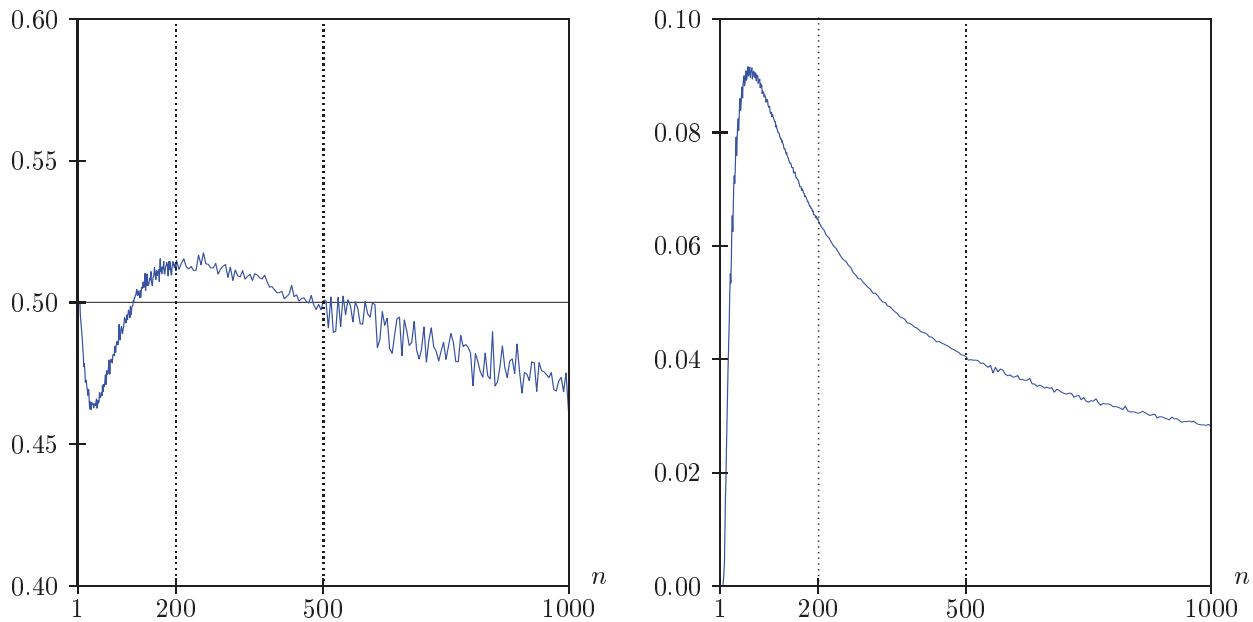


Abbildung 2.24: **(Rucksack-Problem)** Erfolgskurve (links) und Kurve der Schrittzahlquotienten (rechts) von Mehrheitsbildung in Abhängigkeit von n unter Gruppengröße 3 und probabilistisch wiederholter zyklischer schneller lokaler Suche

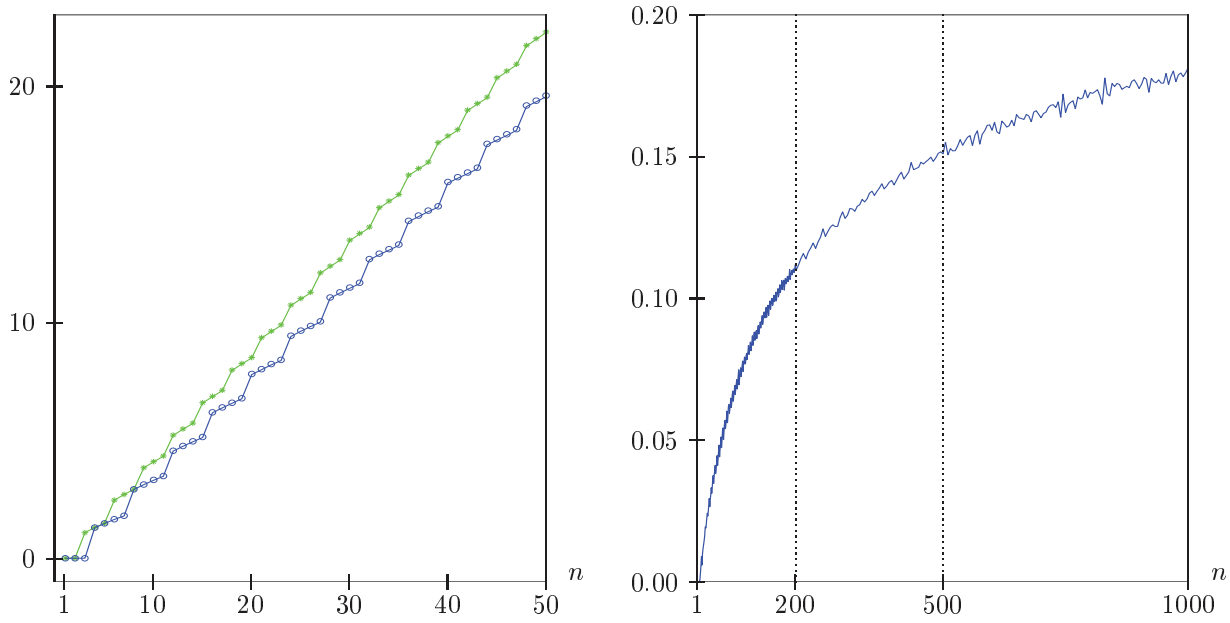
$n = 205, 210, 215, \dots, 500$ wurden dazu je 100 000, für $n = 505, 510, 515, \dots, 1000$ je 10 000 Instanzen gebildet. Wie schon bei allen in den vorangegangenen Abschnitten behandelten Problemen ist auch hier ein Misserfolgstal zu erkennen. Für größere n steigen die Erfolgsquoten allerdings nur geringfügig an und fallen dann wieder ab. Die Kurve der Schrittzahlquotienten zeigt im Wesentlichen (d.h. in ihrer Grobstruktur) einen unimodalen Verlauf. Alle Schrittzahlquotienten sind deutlich kleiner als 1.

Die blaue „Kurve“ in Abbildung 2.25 links zeigt schließlich die durchschnittlichen normierten Zielfunktionswerte $\bar{f}_{\text{zf}}(n)$ der in den zufälligen Läufen lokaler Suche gewonnenen lokal maximalen Lösungen. Zur Normierung wurden die f -Werte durch die Maximalgröße W geteilt. Die Durchschnittswerte $\bar{f}_{\text{Mh}}(n)$ nach den Mehrheitsläufen sind als blaue Kreise eingezeichnet. Die sichtbare „Bündelung“ der Werte in Vierergruppen ist auf die spezielle Wahl der Rucksackgröße $S = \lfloor \frac{n}{4} \rfloor \cdot W$ zurückzuführen. Zum Vergleich sind analoge Berechnungen für $S = \lfloor \frac{n}{3} \rfloor \cdot W$ durchgeführt worden. Die zugehörige „Kurve“ mit entsprechenden Sternen ist in grüner Farbe skizziert. Hier treten die Werte in Dreiergruppen „gebündelt“ auf.

Da die Werte $\bar{f}_{\text{Mh}}(n)$ und $\bar{f}_{\text{zf}}(n)$ in der Graphik optisch nicht unterscheidbar sind, sind zusätzlich in Abbildung 2.25 rechts die Differenzen $\bar{f}_{\text{Mh}}(n) - \bar{f}_{\text{zf}}(n)$ dargestellt. Es zeigt sich, dass $\bar{f}_{\text{Mh}}(n) \geq \bar{f}_{\text{zf}}(n)$ für alle n gilt. Die Differenzen wachsen mit n im Wesentlichen (d.h. in ihrer Grobstruktur) streng monoton, jedoch nur *sublinear*, d.h.

$$\lim_{n \rightarrow \infty} \frac{|\bar{f}_{\text{Mh}}(n) - \bar{f}_{\text{zf}}(n)|}{n} = 0. \quad (\star)$$

Bei allen Problemen aus 2.2.1 bis 2.2.4 lässt sich dagegen ein *lineares* Wachstum der

Abbildung 2.25: **(Rucksack-Problem)**

links: $\bar{f}_{zf}(n)$ bzw. $\bar{f}_{Mh}(n)$ (Rucksackgröße $S = \lfloor \frac{n}{4} \rfloor \cdot W$ (blau) bzw. $S = \lfloor \frac{n}{3} \rfloor \cdot W$ (grün))

rechts: $\bar{f}_{Mh}(n) - \bar{f}_{zf}(n)$

entsprechenden Differenzen in Abhängigkeit der Zahl der Variablen beobachten: Hier sind die zu (\star) analogen Grenzwerte stets endlich und echt größer als Null. Diese Feststellung lässt sich unter jeweils allen untersuchten Gruppengrößen und Varianten lokaler Suche treffen. An dieser Stelle wird auf die Darstellung der entsprechenden Kurven verzichtet.

2.2.6 Erfolgsquoten für das Counting-Ones-Problem

Beim Counting-Ones-Problem ist eine Lösung $x = (x_1, \dots, x_n) \in \{0, 1\}^n$ mit $n \geq 1$ gesucht, welche in ihren Komponenten möglichst viele Einsen hat, d.h. für die

$$f(x) = \#\{i : x_i = 1\}$$

größtmöglich ist. Dieses „Problem“ hat den Vorteil, dass Lösungen einer fest vorgegebenen Güte auf einfache Weise konstruiert werden können. Insbesondere wird kein Verfahren lokaler Suche benötigt. Am Counting-Ones-Problem lässt sich daher untersuchen, inwieweit die lokale Suche, gleich welcher Variante, überhaupt für die Konvergenz des Erfolgs von Mehrheitsbildung gegen 1 notwendig ist. Dazu werden speziell für dieses Problem die *p-Erfolgsquoten* definiert:

DEFINITION 2.8 (*p*-Erfolgsquote)

Sei $p \in [0, 1]$. Weiter seien $x^j(b) = (x_1^j(b), x_2^j(b), \dots, x_n^j(b)) \in \{0, 1\}^n$ für $j = 1, 2, \dots, 7$ und $b = 1, 2, \dots, 100\,000$ wie folgt gewählt:

$$x_i^j(b) = \begin{cases} 1 & \text{mit Wahrscheinlichkeit } p \\ 0 & \text{mit Wahrscheinlichkeit } 1 - p \end{cases} \quad \text{für } i = 1, \dots, n$$

Für $b = 1, 2, \dots, 100\,000$ sei $x^8(b)$ der Mehrheitsvektor zu $x^5(b), x^6(b), x^7(b)$ und

$$\begin{aligned}\max_{\text{zf}}(b) &:= \max \{f(x^j(b)) : j = 1, \dots, 4\}, \\ \max_{\text{Mh}}(b) &:= \max \{f(x^j(b)) : j = 5, \dots, 8\}.\end{aligned}$$

Dann heißt

$$\frac{\#\{b : \max_{\text{Mh}}(b) > \max_{\text{zf}}(b)\} + \frac{1}{2} \#\{b : \max_{\text{Mh}}(b) = \max_{\text{zf}}(b)\}}{100\,000}$$

die (empirische) **p -Erfolgsquote** von Mehrheitsbildung für das Counting-Ones-Problem unter n (und Gruppengröße 3). Die graphische Veranschaulichung der Folge der p -Erfolgsquoten unter n mit verbundenen Folgengliedern heißt eine (empirische) **p -Erfolgskurve**.

Abbildung 2.26 zeigt die p -Erfolgskurven für $p \in \{0.50, 0.55, 0.60, \dots, 0.95\}$ im Bereich $2 \leq n \leq 500$ (links) bzw. $2 \leq n \leq 50$ (rechts). Man erkennt, dass die 0.50-Erfolgskurve kon-

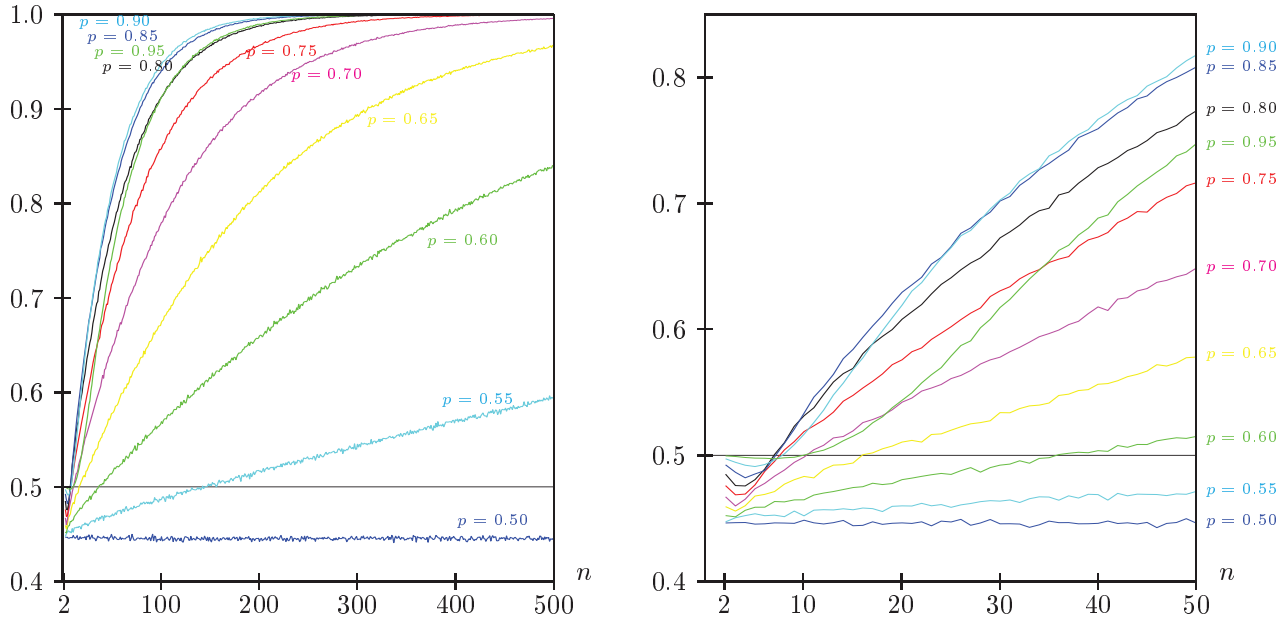


Abbildung 2.26: **(Counting-Ones-Problem)** p -Erfolgskurven von Mehrheitsbildung für verschiedene $p \in [\frac{1}{2}, 1[$

stant ist. Zudem konvergieren alle p -Erfolgskurven mit $p \in]\frac{1}{2}, 1[$ für $n \rightarrow \infty$ gegen 1. Diese Aussage ist zwar aus Abbildung 2.26 nicht eindeutig ersichtlich, lässt sich aber theoretisch nachweisen:

PROPOSITION 2.9

Seien $n \geq 1$ und $f : \{0, 1\}^n \rightarrow \mathbb{R}$. Weiter sei $(\Omega, 2^\Omega, P)$ ein diskreter Wahrscheinlichkeitsraum und $X = (X_1, X_2, \dots, X_n) : \Omega \rightarrow \{0, 1\}^n$ ein quadratisch P -summierbarer Zufallsvektor. Für $j = 1, 2, \dots, 7$ seien $X^j = (X_1^j, \dots, X_n^j)$ unabhängig und identisch wie X verteilt und $X^8 = (X_1^8, \dots, X_n^8)$ definiert durch

$$X_i^8 := \begin{cases} 1, & \text{falls } X_i^5 + X_i^6 + X_i^7 \geq 2 \\ 0, & \text{falls } X_i^5 + X_i^6 + X_i^7 \leq 1 \end{cases}$$

für $i = 1, \dots, n$. Ferner sei $\Delta_f(n) := f(X^8) - f(X)$. Für die Erwartung $E[\Delta_f(n)]$ sowie die Varianz $V[\Delta_f(n)]$ gelte:

$$(i) \quad E[\Delta_f(n)] > 0 \quad \text{für } n = 1, 2, \dots$$

$$(ii) \quad \lim_{n \rightarrow \infty} \frac{V[\Delta_f(n)]}{(E[\Delta_f(n)])^2} = 0$$

Schließlich seien

$$\begin{aligned} \max_{\text{zf}}(n) &:= \max \{f(X^j) : j = 1, \dots, 4\}, \\ \max_{\text{Mh}}(n) &:= \max \{f(X^j) : j = 5, \dots, 8\} \end{aligned}$$

und

$$Y(n) := \begin{cases} 1, & \text{falls } \max_{\text{Mh}}(n) > \max_{\text{zf}}(n) \\ \frac{1}{2}, & \text{falls } \max_{\text{Mh}}(n) = \max_{\text{zf}}(n) \\ 0, & \text{falls } \max_{\text{Mh}}(n) < \max_{\text{zf}}(n) \end{cases}.$$

Dann gilt für die Erwartungen von $Y(n)$, $n = 1, 2, \dots$:

$$\lim_{n \rightarrow \infty} E[Y(n)] = 1$$

Der Beweis erfolgt mit Hilfe der Ungleichung von TSCHEBYSCHEFF. □

DEFINITION 2.10 (p -Erfolg)

Unter den Bezeichnungen von Proposition 2.9 sei $f(x_1, \dots, x_n) := \#\{i : x_i = 1\}$ für alle $(x_1, \dots, x_n) \in \{0, 1\}^n$ und $P[X_i = 1] = p$ für alle $i \in \{1, \dots, n\}$ und ein festes $p \in [0, 1]$. Dann heißt $E_p(n) := E[Y(n)]$ der **p -Erfolg** von Mehrheitsbildung für das Counting-Ones-Problem unter n (und Gruppengröße 3).

Es ist klar, dass die empirische p -Erfolgsquote von Mehrheitsbildung für das Counting-Ones-Problem eine Näherung des p -Erfolgs unter jedem n liefert. Ferner gilt:

SATZ 2.11

Für jedes $p \in]\frac{1}{2}, 1[$ konvergiert der p -Erfolg von Mehrheitsbildung für das Counting-Ones-Problem für $n \rightarrow \infty$ gegen 1.

BEWEIS. $f(X)$ ist binomialverteilt mit Parametern n und p . Also gilt $E[f(X)] = np$ und $V[f(X)] = np(1-p)$. Wegen $P[X_i^8 = 1] = p^3 + 3p^2(1-p) =: p'$ für $i = 1, \dots, n$ ist auch $f(X^8)$ binomialverteilt mit Parametern n und p' , und es gilt $E[f(X^8)] = np'$ sowie $V[f(X^8)] = np'(1-p')$. Daraus folgt

$$E[\Delta_f(n)] = np' - np = n \cdot (p^3 + 3p^2(1-p) - p) = n \cdot 2p(p - \frac{1}{2})(1-p) > 0.$$

Da $f(X)$ und $f(X^8)$ unabhängig sind, gilt ferner $V[\Delta_f(n)] = V[f(X^8)] - V[f(X)]$ und somit

$$\frac{V[\Delta_f(n)]}{(E[\Delta_f(n)])^2} = \frac{n(p'(1-p') - p(1-p))}{n^2(p' - p)^2} \xrightarrow{n \rightarrow \infty} 0.$$

Nach Proposition 2.9 folgt $\lim_{n \rightarrow \infty} E_p(n) = 1$. □

Satz 2.11 zeigt, dass für das erfolgreiche Abschneiden von Mehrheitsbildung kein (spezielles) lokales Suchverfahren notwendig ist. Vielmehr müssen die zur Mehrheitsbildung herangezogenen Lösungen eine gewisse überdurchschnittliche Güte hinsichtlich der Zielfunktion mitbringen. Wahrscheinlich lässt sich die lokale Suche in Definition 2.1 auch durch moderne Heuristiken wie *Simulated Annealing* ([26],[9]) oder *Tabusuche* ([18],[19]) ersetzen, ohne den „Erfolg“ von Mehrheitsbildung zu schmälern. Mehrheitsbildung sollte nicht als „starrer“ Algorithmus in Verbindung mit einer festen Heuristik verstanden werden. Vielmehr ist Mehrheitsbildung eine Meta-Heuristik, die ihre Wirkung zusammen mit guten heuristischen Verfahren entfaltet.

Darüber hinaus ist auch die Ausbildung von **Misserfolgstälern** ohne lokale Suche möglich. Alle in Abbildung 2.26 dargestellten p -Erfolgskurven mit $p > 0.50$ weisen ein Misserfolgstal auf, welches mit wachsendem p flacher wird. Ferner existiert ein $p_0 \in]\frac{1}{2}, 1[$, so dass die Misserfolgstäler mit wachsendem p für $p \in]\frac{1}{2}, p_0]$ kürzer und für $p \in [p_0, 1[$ wieder länger werden. Dies sollte nicht weiter verwundern, da die 1-Erfolgskurve trivialerweise konstanten Wert 0.5 hat.

Wie im Beweis von Satz 2.11 gesehen, haben Mehrheitsvektoren zu binomial mit Parametern n und p verteilten Zufallsvektoren im Durchschnitt bessere f -Werte als die Zufallsvektoren selbst. Dies gilt für alle $p \in]\frac{1}{2}, 1[$ und alle $n \geq 1$, also insbesondere für kleine n . Die Existenz der Misserfolgstäler erscheint deshalb paradox. Eine mögliche Erklärung ist die folgende: Seien $p \in]\frac{1}{2}, 1[$ und $n \geq 1$. Dann gilt mit den Bezeichnungen von Satz 2.11 $p' > p > \frac{1}{2}$ und folglich

$$\begin{aligned} V[f(X)] - V[f(X^8)] &= n(p(1-p) - p'(1-p')) \\ &= n(p' - p)(p' + p - 1) \\ &> 0. \end{aligned}$$

Die f -Werte der Zufallsvektoren streuen also um ihren Erwartungswert $E[f(X)]$ stärker als die f -Werte der Mehrheitsvektoren um $E[f(X^8)]$. Dieser Unterschied im Streuverhalten reicht bei kleinen n dazu aus, dass die Wahrscheinlichkeit für das Ereignis $\max_{zf} > \max_{Mh}$ (vgl. Proposition 2.9) größer als $\frac{1}{2}$ ist. Der p -Erfolg von Mehrheitsbildung ist dann kleiner als $\frac{1}{2}$. Bei großen n wird dieser Effekt durch die gewachsene Erwartungswertdifferenz $E[f(X^8)] - E[f(X)]$ ausgeglichen.

Mittels einer analogen Argumentation werden im folgenden Abschnitt auch die Misserfolgstäler der Erfolgskurven bei den vorangegangenen Optimierungsproblemen erklärt.

2.3 Diskussion der Misserfolgstäler

Die ersten Versuche zum Einsatz von Mehrheitsbildung für die vorliegende Dissertation wurden am Problem Max-3-SAT durchgeführt. Die Ausgangserwartung war, dass sich durch Mehrheitsläufe bessere Zielfunktionswerte als durch zufällige Läufe erreichen lassen. Zunächst wurden zufällige Instanzen erzeugt und zu jeder Instanz mehrere Blöcke zu je drei zufälligen Läufen und einem Mehrheitslauf gestartet. Nach jedem Block wurde

der Zielfunktionswert f^{Mh} des Mehrheitslaufs den Werten f^1, f^2, f^3 der vorangegangenen zufälligen Läufe gegenübergestellt. Schon bald war klar, dass f^{Mh} typischerweise im Intervall $[f_{\min}, f_{\max}]$ mit $f_{\min} = \min \{f^1, f^2, f^3\}$ und $f_{\max} = \max \{f^1, f^2, f^3\}$ liegt, manchmal größer als f_{\max} und sehr selten kleiner als f_{\min} ist. Im Bemühen um eine systematische Untersuchung des „Erfolgs“ von Mehrheitsbildung wurden die Erfolgsquoten definiert. Diese basierten allerdings zunächst nicht auf der Dimension n des Lösungsraums, sondern jeweils auf einer fest gewählten Problemistanz. Gemittelt wurde über mehrere hundert Viererblöcke von Läufen lokaler Suche für dieselbe Instanz. Als Anzahl n von Variablen wurde stets an 10, 20 oder 30 festgehalten. Die Ergebnisse waren enttäuschend: Für nahezu alle Instanzen waren die Erfolgsquoten kleiner als $\frac{1}{2}$. Auf der Suche nach erfolgreichen Beispielen wurde deshalb von den zufällig erzeugten Instanzen zu „praxistypischen“ Benchmarks gewechselt. Das Ergebnis waren die in Abschnitt 2.1 vorgestellten Daten, insbesondere für „Hanoi 5“. Sie bildeten den Durchbruch zu einer Erfolgsserie der Mehrheitsbildung. Zunächst lag der Verdacht nahe, dass Mehrheitsbildung nur bei Instanzen speziell gewählter Struktur zum Erfolg führt. Erst lange Zeit später, nach Berechnen der Erfolgsquoten gemäß Definition 2.1, stellte sich heraus, dass vielmehr die zu geringe Problemgröße für den anfänglichen Misserfolg verantwortlich gewesen war.

Die Erfolgskurven aller in Abschnitt 2.2 behandelten Optimierungsprobleme mit Lösungsraum $\{0, 1\}^n$ weisen ein Misserfolgstal auf: In einem Intervall kleiner Problemparameter n sind die Erfolgsquoten höchstens $\frac{1}{2}$. Dies erscheint zunächst paradox, da die Zielfunktionswerte lokal minimaler Lösungen aus Mehrheitsläufen im Durchschnitt für kein einziges n schlechter sind als die entsprechenden Werte aus zufälligen Läufen. Kurze Misserfolgstäler wie etwa beim Ising-Spincas-Problem (vgl. Abschnitt 2.2.4) sind für die Praxis harmlos. Das Dynamische Optimierungsproblem weist dagegen Misserfolgstäler bis hin zu knapp 500 Variablen auf (vgl. Abschnitt 2.2.2).

Im Folgenden werden einige weiterführende Beobachtungen rund um Misserfolgstäler zur Diskussion gestellt. Zum besseren Verständnis der ergänzenden Bemerkungen wird vorab festgehalten, dass Mehrheitsbildung im Lösungsraum $\{0, 1\}^n$ mit der HAMMING-Nachbarschaft der Medianbildung entspricht:

BEMERKUNG 2.12

Seien $n \geq 1$, $L \geq 2$ und x^1, x^2, \dots, x^L , $x \in \{0, 1\}^n$. Dann ist x ein Mehrheitsvektor zu x^1, \dots, x^L genau dann, wenn x ein **Median** zu x^1, \dots, x^L ist, d.h. wenn für alle $v \in \{0, 1\}^n$ gilt:

$$\sum_{\ell=1}^L d_H(x, x^\ell) \leq \sum_{\ell=1}^L d_H(v, x^\ell)$$

$d_H(y, z)$ bezeichne dabei den HAMMING-Abstand von $y, z \in \{0, 1\}^n$, d.h. die Zahl an Komponenten, in denen sich y und z unterscheiden.

Der Beweis erfolgt auf elementare Weise durch Nachrechnen. □

2.3.1 Misserfolgstäler bei modifizierten Erfolgskurven

Bei Minimierungsaufgaben mit Lösungsraum $\{0, 1\}^n$ wächst die Gesamtzahl der Lösungen exponentiell mit n . Typischerweise gilt dies auch für die durchschnittliche Anzahl *lokal* minimaler Lösungen. Betrachte etwa das Shiftregister-Problem, dessen Instanzen zum Parameter $n \geq 4$ die dreielementigen Teilmengen von $\{1, \dots, n-1\}$ sind (vgl. Abschnitt 2.2.3). In Tabelle 2.6 sind alle $n \in \{4, 5, 6, \dots, 25\}$ und zu jedem n die durchschnittliche Anzahl $\overline{A_1}(n)$ der lokal minimalen Lösungen sowie die durchschnittliche Anzahl $\overline{A_2}(n)$ der Äquivalenzklassen lokal minimaler Lösungen dargestellt. Der zugrunde gelegte Äquivalenzbegriff ist in Abschnitt 2.2.3 eingeführt worden. Gemittelt wurde jeweils über die $\binom{n-1}{3}$ Instanzen zum Parameter n . Durch graphische Veranschaulichung der Werte lässt sich beobachten, dass $\overline{A_i}(n)$ für $i = 1, 2$ asymptotisch wie α_i^n für ein $\alpha_i > 1$ wächst.

n	4	5	6	7	8	9	10	11
$\overline{A_1}(n)$	6.00	10.00	17.60	30.80	36.34	59.79	91.67	122.83
$\overline{A_2}(n)$	2.00	1.00	2.50	2.20	3.77	3.46	5.95	5.58
n	12	13	14	15	16	17	18	19
$\overline{A_1}(n)$	201.36	294.04	498.67	699.23	1044.71	1600.91	2538.76	3634.87
$\overline{A_2}(n)$	10.74	11.31	19.97	23.79	36.40	47.09	74.85	95.65
n	20	21	22	23	24	25		
$\overline{A_1}(n)$	5376.02	8996.77	12359.95	18847.51	31652.13	42746.34		
$\overline{A_2}(n)$	140.56	215.20	287.48	409.73	672.65	855.27		

Tabelle 2.6: **(Shiftregister-Problem)** Durchschnittliche Anzahlen $\overline{A_1}(n)$ bzw. $\overline{A_2}(n)$ paarweise verschiedener bzw. paarweise nicht-äquivalenter lokal minimaler Lösungen, jeweils gemittelt über alle Instanzen zum Parameter n

Sind nun $y^1, y^2, y^3 \in \{0, 1\}^n$ lokal minimale Lösungen, so ist bei kleinen n aufgrund der geringen Gesamtzahl lokal minimaler Lösungen die Wahrscheinlichkeit groß, dass die Vektoren nicht paarweise verschieden sind. Gilt $y^i = y^j =: y$ für $1 \leq i < j \leq 3$, so stimmt auch der Mehrheitsvektor y^{Mh} zu y^1, y^2, y^3 mit y überein. Ein Lauf lokaler Suche mit Startvektor y^{Mh} „fällt“ dann „aus“. Ein Lauf mit zufälligem Startvektor hat dagegen die Chance, zu einer neuen lokal minimalen Lösung zu führen. Dieser Effekt drückt die empirische Erfolgsquote entsprechend nach unten.

Die Stichhaltigkeit dieser Argumentation ist am Beispiel des Dynamischen Optimierungsproblems genauer geprüft worden. Betrachte zunächst die blaue „Kurve“ in Abbildung 2.27 links. Hierbei handelt es sich um die aus Abbildung 2.15 bekannte Erfolgskurve von Mehrheitsbildung unter Gruppengröße 3 und probabilistisch wiederholter zyklischer schneller lokaler Suche. Zu jedem Parameter $n \in \{2, 3, \dots, 1000\}$ wurden 100 000 Instanzen zufällig und unabhängig gewählt. Pro Instanz wurde der größte f -Wert aus vier zufälligen Läufen (Läufe 1,2,3,4) mit dem größten f -Wert aus drei zufälligen Läufen (Läufe 5,6,7) und einem Mehrheitslauf (Lauf 8) verglichen.

In einem zweiten Experiment wurde unter denselben Rahmenbedingungen eine Problem-Instanz verworfen, wenn der Startvektor des Mehrheitslaufs mit einer der lokal minimalen

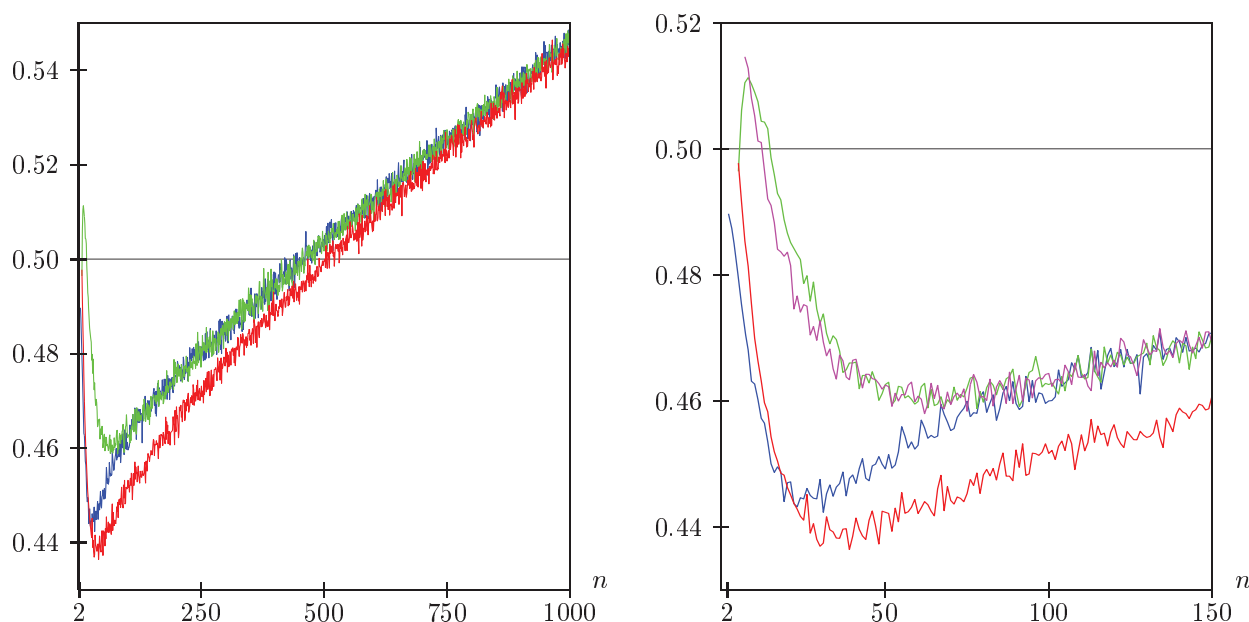


Abbildung 2.27: **(Dynamisches Optimierungsproblem)** „Erfolgskurven“ von Mehrheitsbildung in Abhängigkeit von n unter Gruppengröße 3 und probabilistisch wiederholter zyklischer schneller lokaler Suche

Lösungen aus den Läufen 5,6,7 übereinstimmte. In diesem Fall wurde die Instanz durch eine neue, zufällig gewählte Instanz ersetzt. Die so erhaltene „Erfolgskurve“ ist mit grüner Farbe skizziert. Man erkennt, dass das Misserfolgstal deutlich kleiner geworden ist. Die in Abbildung 2.27 rechts dargestellten „Kurven“ sind die Einschränkungen der links gezeigten „Kurven“ entsprechender Farbe auf den Parameterbereich $n \leq 150$. Man beachte, dass die grüne „Kurve“ über die 0.50-Hürde ansteigt, dann wieder fällt und ab einem bestimmten Parameter n_0 ziemlich genau mit der blauen „Kurve“ zusammenfällt. Für $n > n_0$ spielt der beschriebene Effekt also praktisch keine Rolle mehr.

In der Hoffnung, das Misserfolgstal ganz weg zu bekommen, wurde nun ein Instanz genau dann verworfen, wenn der Mehrheitsvektor eine lokal minimale Lösung war. Diese Regel schließt den vorangegangenen Fall mit ein. Das Ergebnis zeigt die rote „Kurve“. Überraschenderweise ist das zugehörige Misserfolgstal länger und tiefer als obige Täler. Die Möglichkeit zu verbieten, allein durch Mehrheitsbildung eine neue lokal minimale Lösung zu erhalten, bildet also eine zu starke Einschränkung. Umgekehrt zeigt diese Beobachtung, dass auch die bloße Mehrheitsbildung, d.h. ohne anschließende heuristische Verfahren, in einigen Fällen zu neuen, besseren Lösungen führt.

Betrachte schließlich die magentafarbene „Erfolgskurve“ in der Abbildung rechts. Hier wurde eine Instanz verworfen, falls Lauf 8 zu einer lokal minimalen Lösung führte, die mit einer der lokal minimalen Lösungen aus den Läufen 5,6,7 übereinstimmte. Um den Block der ersten vier zufälligen Läufe nicht zu benachteiligen, wurde die Instanz ebenso verworfen, falls Lauf 4 in einer der lokal minimalen Lösungen aus den Läufen 1,2,3 endete. In der linken Graphik wurde auf die Darstellung dieser „Erfolgskurve“ verzichtet, da sie von der grünen „Kurve“ optisch nicht unterscheidbar ist und diese überdecken würde.

Für das Misserfolgstal sind also mehrere Effekte verantwortlich:

1. Die Erfolgskurve der Mehrheitsbildung (blaue „Kurve“ in Abbildung 2.27) weist an einer Stelle n_0 eine „Wendestelle“ auf. Für $n < n_0$ trägt folgender Effekt wesentlich zum Misserfolgstal bei: Sei $y^{\text{Mh}} \in \{0, 1\}^n$ der Mehrheitsvektor zu drei unabhängig mittels zufälligen Läufen lokaler Suche gefundenen lokal minimalen Lösungen $y^1, y^2, y^3 \in \{0, 1\}^n$. Weiter sei $x \in \{0, 1\}^n$ zufällig gewählt. Dann ist die Wahrscheinlichkeit, dass ein Lauf mit Startvektor y^{Mh} zu einer der Lösungen y^1, y^2, y^3 führt, größer als die Wahrscheinlichkeit, dass dies ausgehend von x passiert. Für $n \geq n_0$ spielt dieser Effekt eine eher untergeordnete Rolle.
2. Aus Bemerkung 2.12 folgt insbesondere, dass wiederholte lokale Suche mit ausschließlich zufälligen Läufen „den Lösungsraum gleichmäßiger abdeckt“ als wiederholte lokale Suche unter Einsatz von Mehrheitsläufen. „Gleichmäßiger“ heißt für kleine n auch „besser“. Dies gilt auch dann, wenn nur Mehrheitsläufe zugelassen sind, die zu neuen Lösungen führen (magentafarbene „Kurve“ in Abbildung 2.27). Eine mögliche Erklärung hierfür ist folgende:
Man erinnere sich an die Häufigkeitsverteilungen der lokalen Maxima im Max-SAT-Beispiel „Hanoi 5“ aus Abschnitt 2.1.1. Die rote „Glocke“ der Mehrheitsläufe lag unter beliebiger Variante lokaler Suche deutlich rechts von der schwarzen „Glocke“ der zufälligen Läufe. Vermutlich „wandert“ die „Glocke“ der Mehrheitsläufe bei zufälligen Instanzen typischerweise mit wachsendem Problemparameter aus der „Glocke“ der zufälligen Läufe „heraus“. Bei zu kleinen Parametern kann die Erfolgsquote daher trotz besserer durchschnittlicher Zielfunktionswerte kleiner als 0.5 ausfallen. Hier ist die Streuung der f -Werte der lokal maximalen Lösungen aus den Mehrheitsläufen um ihren Mittelwert zu gering.

2.3.2 Erfolg und Misserfolg bei Pool-Erfolgskurven

In die Ermittlung einer Erfolgsquote geht pro Problemistanz nur ein einziger Mehrheitslauf ein. Womöglich würde der Erfolg von Mehrheitsbildung auch bei kleinen Problemparametern höher ausfallen, wenn der Anteil an Mehrheitsläufen größer wäre. Die Idee ist nun, 2ϱ zufällige Läufe (Block I) mit ϱ zufälligen Läufen und ϱ Mehrheitsläufen (Block II) zu vergleichen ($\varrho \geq 1$). Für Block II wird dazu mittels wiederholter lokaler Suche ein **Pool** aus ϱ lokal minimalen Lösungen $x^1, x^2, \dots, x^\varrho \in \{0, 1\}^n$ ermittelt. Aus diesen werden auf bestimmte Weise ϱ Tripel (x^i, x^j, x^k) mit $1 \leq i \leq j \leq k \leq \varrho$ und zu jedem Tripel der Mehrheitsvektor gebildet. Anschließend finden ϱ Mehrheitsläufe mit den erhaltenen Mehrheitsvektoren als Startlösungen statt. Eine detaillierte Beschreibung des Konstruktionsverfahrens für die ϱ Lösungstriple wird in Kapitel 4.2, S. 91, nachgeholt. Die Triple sollten jedenfalls „so verschieden wie möglich“ sein. Innerhalb eines Tripels sollten gleiche Lösungen vermieden werden (vgl. Abschnitt 2.3.1).

Die (empirische) **Pool-Erfolgsquote** für einen Problemparameter n wird analog zur Erfolgsquote unter n gemäß Definition 2.1 gebildet. Für festes n wird die graphische Veranschaulichung der Folge der Pool-Erfolgsquoten in Abhängigkeit der Poolgröße ϱ mit verbundenen Gliedern als **Pool-Erfolgskurve** bezeichnet. In Abbildung 2.28 sind ver-

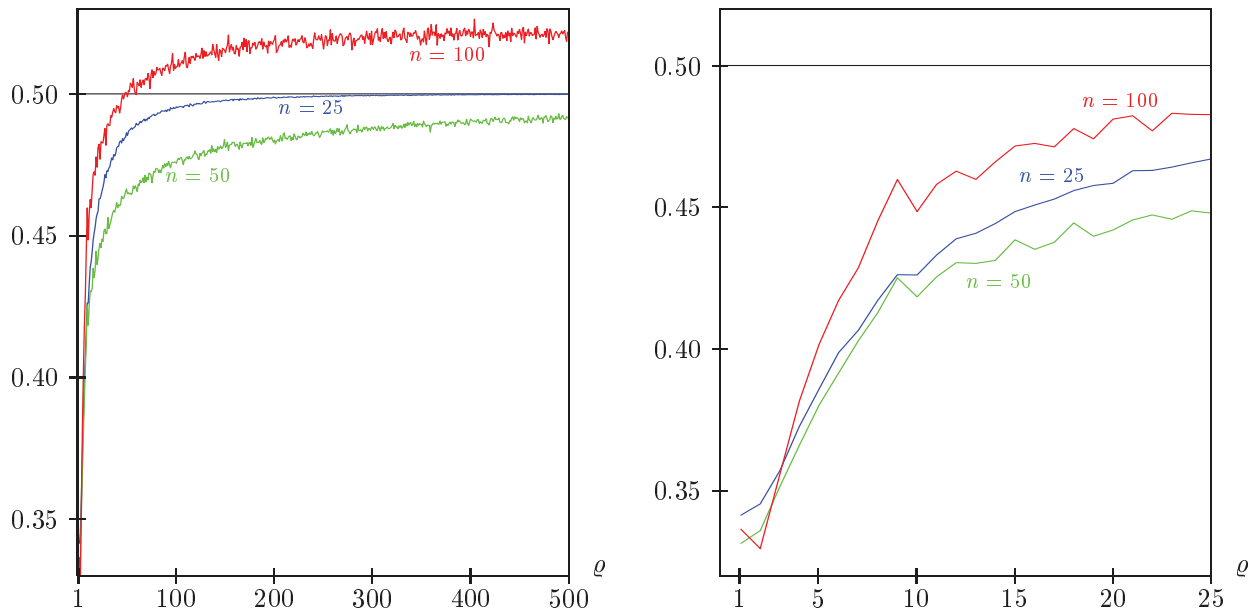


Abbildung 2.28: **(Dynamisches Optimierungsproblem)** Pool-Erfolgskurven unter Gruppengröße 3 und probabilistisch wiederholter zyklischer schneller lokaler Suche

schiedene Pool-Erfolgskurven für das Dynamische Optimierungsproblem dargestellt, links im Bereich $1 \leq \varrho \leq 500$, rechts im Ausschnittsbereich $1 \leq \varrho \leq 25$. Jede Pool-Erfolgsquote wurde auf der Basis von 100 000 unabhängigen zufälligen Problem instanzen gebildet. Als Heuristik wurde die probabilistisch wiederholte zyklische schnelle lokale Suche eingesetzt. Die „Kurve“ zu $n = 25$ ist blau, zu $n = 50$ grün und zu $n = 100$ rot skizziert. Alle drei Parameter gehören dem Misserfolgstal von Mehrheitsbildung unter Gruppengröße 3 an (siehe Abbildung 2.15, S. 52). Trotzdem überschreitet die „Kurve“ zu $n = 100$ die 0.5-Hürde ab einem gewissen ϱ , während die beiden anderen „Kurven“ darunter bleiben. Der Parameter 100 scheint sich also von 25 und 50 signifikant zu unterscheiden. Vermutlich existiert ein Schwellparameter n^* , so dass die Pool-Erfolgskurven für $n < n^*$ gegen einen Wert von 0.5 oder kleiner konvergieren und für $n > n^*$ mit Grenzwert größer als 0.5.

Kapitel III

Randomisierte Mehrheitsbildungen

In diesem Kapitel werden Varianten der Mehrheitsbildung diskutiert. Insbesondere werden Konvergenzeigenschaften ihres Erfolgs untersucht. Bei Problemparametern des Misserfolgstals von Mehrheitsbildung werden alternative Bildungsregeln vorgeschlagen.

3.1 Einstimmigkeit, Proporz, Antimehrheit

Die Ausbildung von Misserfolgstätern bei den Erfolgskurven von Mehrheitsbildung zeigt, dass bei kleinen Problemparametern auf Mehrheitsbildung besser verzichtet werden sollte. Wiederholte lokale Suche mit ausschließlich zufälligen Läufen deckt den Lösungsraum hier deutlich besser ab als bei Einsatz von Mehrheitsläufen. Die Frage ist nun, ob es sinnvoll ist, einen Kompromiss zwischen zufälligen Startvektoren für lokale Suche und Mehrheitsvektoren in der folgenden Weise zu schließen: Eine 0 bzw. 1 wird in einer Komponente nur dann sicher übernommen, wenn der entsprechende Eintrag „einstimmig“ in *allen* Ausgangsvektoren in dieser Komponente auftritt. Unentschiedene Komponenten im Sinne dieser „Einstimmigkeit“ werden je mit Wahrscheinlichkeit $\frac{1}{2}$ auf 0 bzw. 1 gesetzt. Alternativ dazu kann man sich für einen bestimmten Eintrag auch mit einer Wahrscheinlichkeit entscheiden, die zu seinem Vorkommen in den lokal optimalen Lösungen proportional ist. Unter Parametern des Misserfolgstals bietet sich schließlich auch eine Entscheidung für die Antimehrheit der vorkommenden Einträge an. Diese Möglichkeiten werden nun als Bildungsregeln formuliert:

DEFINITION 3.1

Seien $n \geq 1$, $L \geq 2$ und $x^\ell = (x_1^\ell, x_2^\ell, \dots, x_n^\ell) \in \{0, 1\}^n$ für $\ell = 1, \dots, L$. Weiter werde $x = (x_1, \dots, x_n) \in \{0, 1\}^n$ wie folgt gebildet: Für jedes $i \in \{1, \dots, n\}$ werde $x_i = 1$ mit Wahrscheinlichkeit $P_i(1)$ und $x_i = 0$ mit Wahrscheinlichkeit $P_i(0) = 1 - P_i(1)$ gewählt. $P_i(1)$ sei dabei wie folgt definiert:

$$\begin{aligned}
 \text{(i) Mehrheitsregel:} \quad P_i(1) &= \begin{cases} 1, & \text{falls } x_i^1 + \dots + x_i^L > \frac{L}{2} \\ \frac{1}{2}, & \text{falls } x_i^1 + \dots + x_i^L = \frac{L}{2} \\ 0, & \text{falls } x_i^1 + \dots + x_i^L < \frac{L}{2} \end{cases} \\
 \text{(ii) Einstimmigkeitsregel:} \quad P_i(1) &= \begin{cases} 1, & \text{falls } x_i^1 + \dots + x_i^L = L \\ \frac{1}{2}, & \text{falls } x_i^1 + \dots + x_i^L \in]0, L[\\ 0, & \text{falls } x_i^1 + \dots + x_i^L = 0 \end{cases} \\
 \text{(iii) Proporzregel:} \quad P_i(1) &= \frac{1}{L} (x_i^1 + \dots + x_i^L)
 \end{aligned}$$

$$(iv) \quad \textbf{Antimehrheitsregel:} \quad P_i(1) = \begin{cases} 1, & \text{falls } x_i^1 + \dots + x_i^L < \frac{L}{2} \\ \frac{1}{2}, & \text{falls } x_i^1 + \dots + x_i^L = \frac{L}{2} \\ 0, & \text{falls } x_i^1 + \dots + x_i^L > \frac{L}{2} \end{cases}$$

Die Anwendung der Mehrheitsregel (bzw. Einstimmigkeitsregel bzw. Proporzregel bzw. Antimehrheitsregel) auf x^1, \dots, x^L heißt **Mehrheitsbildung** (bzw. **Einstimmigkeitsbildung** bzw. **Proporzbildung** bzw. **Antimehrheitsbildung**).

Geht x durch Mehrheitsbildung (bzw. Einstimmigkeitsbildung bzw. Proporzbildung bzw. Antimehrheitsbildung) aus x^1, \dots, x^L hervor, soll von einem **Mehrheitsvektor** (bzw. **Einstimmigkeitsvektor** bzw. **Proporzvektor** bzw. **Antimehrheitsvektor**) zu x^1, \dots, x^L gesprochen werden.

Ein Lauf lokaler Suche, der in einem Mehrheitsvektor (bzw. Einstimmigkeitsvektor bzw. Proporzvektor bzw. Antimehrheitsvektor) startet, heißt **Mehrheitslauf** (bzw. **Einstimmigkeitslauf** bzw. **Proporzlauf** bzw. **Antimehrheitslauf**).

Die Begriffe **Erfolg**, **Erfolgsquote**, **Erfolgskurve**, **Schrittzahlquotient** und **Kurve der Schrittzahlquotienten** von Einstimmigkeitsbildung, Proporzbildung und Antimehrheitsbildung werden analog zu den entsprechenden Begriffen von Mehrheitsbildung in den Definitionen 2.2, 2.1, 2.3 und beigefügten Bemerkungen formuliert.

In diesem Abschnitt werden einige experimentelle Resultate vorgestellt. Als erstes Optimierungsproblem wird Max-3-SAT betrachtet. Es sei $n \geq 1$ die Anzahl der Variablen und $m^* := m^*(n) := \lceil 4.25n \rceil$ zu jedem n die Zahl der gebildeten Klauseln. In einer Problemistanz zu den Parametern n und m^* besteht jede Klausel aus genau 3 Literalen. Diese werden unabhängig und zufällig gemäß der Gleichverteilung auf der Menge $\{X_1, X_2, \dots, X_n, \neg X_1, \neg X_2, \dots, \neg X_n\}$ der n Variablen sowie ihrer Negationen gewählt. Die Erfolgskurven von Einstimmigkeitsbildung, Proporzbildung und Antimehrheitsbildung unter dem Parameterpaar (n, m^*) , Gruppengröße 3 und probabilistisch wiederholter zyklischer schneller lokaler Suche sind in Abbildung 3.1 illustriert. Zum Vergleich ist die Erfolgskurve von Mehrheitsbildung aus Abbildung 2.5, Seite 40, beigefügt. Zu jedem Parameter $n \in \{1, 2, \dots, 100\}$ wurden 100 000 zufällige Instanzen unabhängig erzeugt, zu $n \in \{105, 110, 115, \dots, 300\} \cup \{350, 400, 450, \dots, 3000\} \cup \{3100, 3200, 3300, \dots, 5000\}$ je 10 000. Die Graphik in Abbildung 3.1 rechts zeigt den Ausschnitt der linken Graphik für $n \leq 100$.

Man sieht, dass die Erfolgskurven von Einstimmigkeitsbildung und Proporzbildung wie schon von Mehrheitsbildung für $n \rightarrow \infty$ gegen 1 konvergieren. Zudem lassen alle drei „Kurven“ ein Misserfolgstal erkennen. Die Erfolgskurve der Antimehrheitsbildung weist dagegen einen anfänglichen „Erfolgshügel“ auf. Für größere n fällt diese „Kurve“ allerdings unter die 0.5-Marke. Für kleine n ist also die Anwendung der Antimehrheitsregel empfehlenswert. Dies rechtfertigt die Intention, für Parameter des Misserfolgstals der Mehrheitsbildung den Lösungsraum möglichst „gleichmäßig“ abzudecken. Man erinnere sich dazu (Bemerkung 2.12), dass im Graphen mit Knotenmenge $\{0, 1\}^n$ und der durch die HAMMING-Nachbarschaft festgelegten Kantenmenge die Mehrheitsvektoren zu lokal optimalen Lösungen $x^1, x^2, \dots, x^L \in \{0, 1\}^n$ gerade die Mediane sind. Antimehrheitsläufe starten dagegen in einer Lösung, die von den lokal optimalen Lösungen einen *maximalen* durchschnittlichen HAMMING-Abstand haben. Dies ist Inhalt der folgenden Bemerkung:

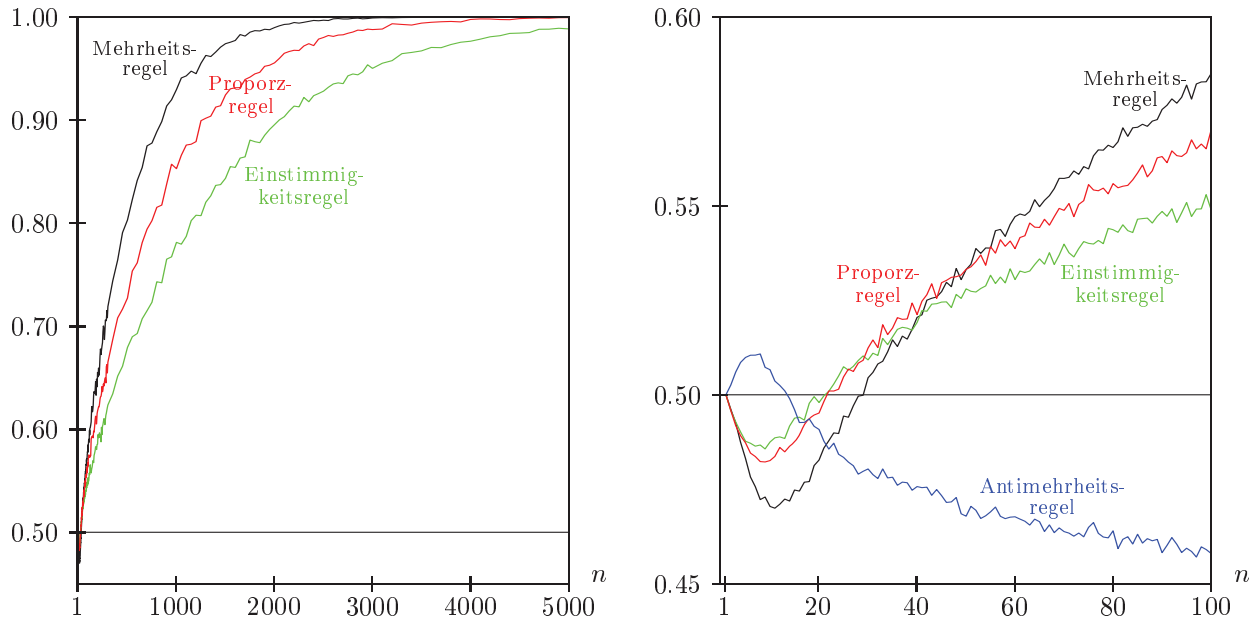


Abbildung 3.1: **(Max-3-SAT)** Erfolgskurven von Mehrheitsbildung, Einstimmigkeitsbildung, Proporzbildung und Antimehrheitsbildung unter (n, m^*) , Gruppengröße 3 und probabilistisch wiederholter zyklischer schneller lokaler Suche in Abhängigkeit von n

BEMERKUNG 3.2

Seien $n \geq 1$, $L \geq 2$ und x^1, x^2, \dots, x^L , $x \in \{0, 1\}^n$. Dann ist x ein Antimehrheitsvektor zu x^1, \dots, x^L genau dann, wenn für alle $v \in \{0, 1\}^n$ gilt:

$$\sum_{\ell=1}^L d_H(x, x^\ell) \geq \sum_{\ell=1}^L d_H(v, x^\ell)$$

Der Beweis ergibt sich unmittelbar aus Bemerkung 2.12. \square

Nach den Bildungsregeln in Definition 3.1 nimmt $|P_i(1) - \frac{1}{2}|$ für jedes $i \in \{1, \dots, n\}$ von der Mehrheitsregel über die Proporzregel hin zur Einstimmigkeitsregel ab. Es ist daher nicht weiter verwunderlich, dass die Erfolgskurve der Proporzbildung in Abbildung 3.1 sowohl hinsichtlich der Tiefe des Misserfolgstals als auch der „Wachstumsgeschwindigkeit“ für große n zwischen den Erfolgskurven von Mehrheitsbildung und Einstimmigkeitsbildung liegt. Ein entsprechendes Ergebnis zeigen die f -Wert-Differenzen in Abbildung 3.2 links: Hier wurden die Differenzen aus den durchschnittlichen f -Werten aus den Mehrheitsläufen (bzw. den Proporz- bzw. Einstimmigkeitsläufen) und den zufälligen Läufen gebildet und in Abhängigkeit von n skizziert. Je höher $|P_i(1) - \frac{1}{2}|$, desto größer ist die entsprechende f -Wert-Differenz. Lokal maximale Lösungen aus Antimehrheitsläufen haben dagegen im Durchschnitt schlechtere f -Werte als zufällige Läufe.

In der rechten Graphik von Abbildung 3.2 sind schließlich die Kurven der Schrittzahlquotienten eingetragen. Die Anwendung der Mehrheits-, Proporz- und Einstimmigkeitsregel für Startvektoren lokaler Suche liefert unter jedem Problemparameter Schrittzahlquotienten kleiner als 1. Von der erst- zur letztgenannten Regel nehmen die Schrittzahlquotienten

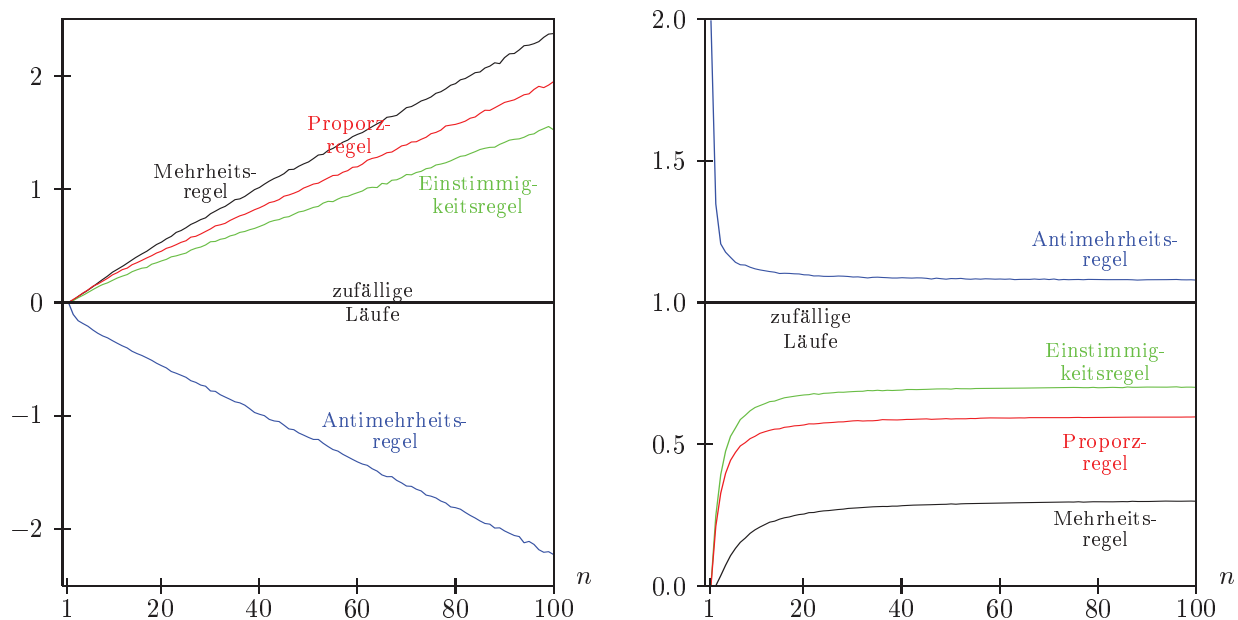


Abbildung 3.2: **(Max-3-SAT)** f -Wert-Differenzen (links) und Kurven der Schrittzahlquotienten (rechts) von Mehrheitsbildung, Einstimmigkeitsbildung, Proporzbildung und Antimehrheitsbildung unter (n, m^*) , Gruppengröße 3 und probabilistisch wiederholter zyklischer schneller lokaler Suche in Abhängigkeit von n

für jedes feste n zu. Antimehrheitsläufe führen dagegen zu Schrittzahlquotienten größer als 1. Alle hier angestellten Beobachtungen sind wegen Definition 3.1 und Bemerkung 3.2 nicht überraschend.

Die Erfolgsquoten, durchschnittlichen f -Wert-Differenzen und Schrittzahlquotienten für die in Definition 3.1 genannten Vektorkonstruktionen sind analog unter wiederholter *voller* lokaler Suche und Gruppengröße 3 ermittelt worden. Ferner wurden diese Größen auch für das SAT-Modell mit „reinen Klauseln“ (vgl. Abschnitt 2.2.1, S. 49) sowie für das Shiftregister-Problem, jeweils unter Proporz- bzw. Einstimmigkeitsbildung, probabilistisch wiederholter zyklischer schneller lokaler Suche und Gruppengröße 3, berechnet. Da sich die erhaltenen „Kurven“ nur unwesentlich von den Graphiken in den Abbildungen 3.1 und 3.2 unterscheiden, wird an dieser Stelle auf ihre Darstellung verzichtet.

Etwas anders sehen die Erfolgskurven für das ± 1 -Ising-Spinglas-Problem auf dem $(n \times n)$ -Torus ohne äußeres Feld aus (Abbildung 3.3). Hier wurden die Erfolgsquoten unter Gruppengröße 3 und probabilistisch bzw. deterministisch wiederholter zyklischer schneller lokaler Suche berechnet. Pro „Kurve“ wurden dazu 100 000 Beispiel-Instanzen zufällig und unabhängig für jedes $n \in \{3, 4, \dots, 200\}$ erzeugt. Man erkennt, dass alle Erfolgskurven für $n \rightarrow \infty$ gegen 1 konvergieren. Zudem weist jede „Kurve“ ein Misserfolgstal auf. Wie für Max-3-SAT oder das Shiftregister-Problem ist auch für das Ising-Spanglas-Problem das Misserfolgstal von Mehrheitsbildung tiefer als von Proporzbildung und letzteres wiederum tiefer als das Misserfolgstal von Einstimmigkeitsbildung. Im Gegensatz zu den oben behandelten Optimierungsproblemen nimmt aber hinsichtlich der Konvergenzgeschwindigkeit gegen 1 nicht die Proporzbildung, sondern die Mehrheitsbildung eine Mittelstellung

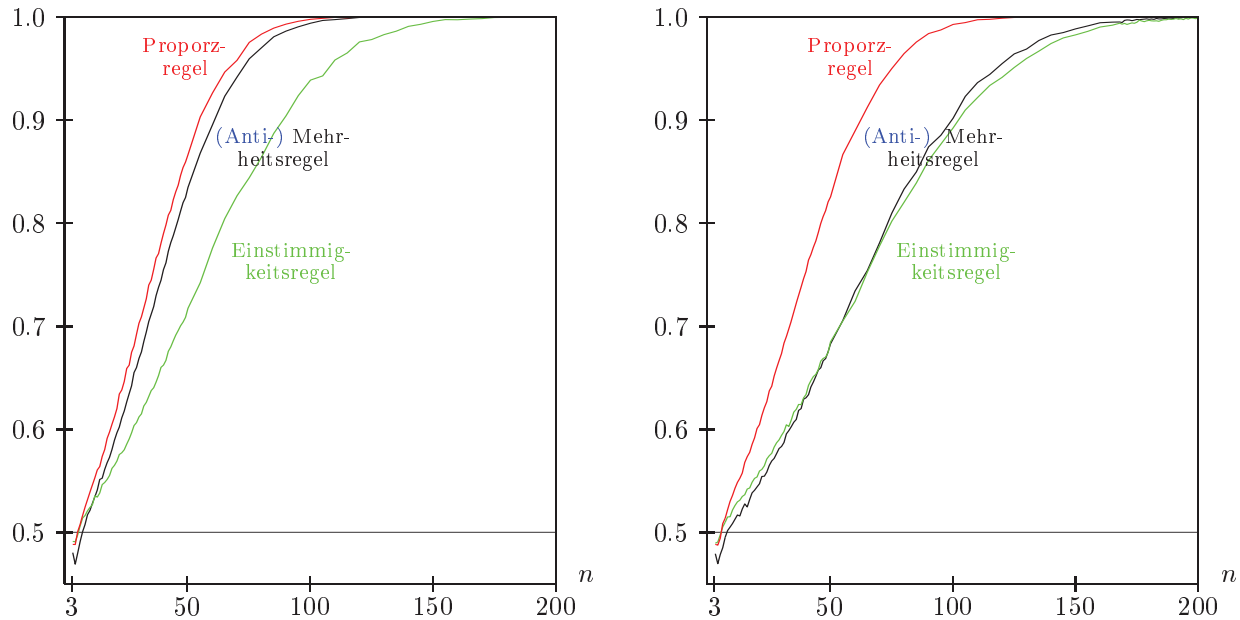


Abbildung 3.3: **(Ising-Spinglas-Problem)** Erfolgskurven von Mehrheitsbildung, Einstimmigkeitsbildung und Proporzbildung in Abhängigkeit von n unter Gruppengröße 3 und probabilistisch (links) bzw. deterministisch (rechts) wiederholter zyklischer schneller lokaler Suche

ein. Die Frage nach einer genauen Begründung für diesen Effekt bleibt an dieser Stelle offen. Vermutlich ist der Effekt auf die Symmetrie der Zielfunktion zurückzuführen: Es gilt $f(x) = f(-x)$ für alle Spinzustände x des Torus mit Elementen aus $\{-1, +1\}$. Die Symmetrie von f stellt darüber hinaus sicher, dass ein durch Anwendung der Mehrheitsregel gewonnener Spinzustand x denselben Zielfunktionswert wie der durch Anwendung der Antimehrheitsregel erhaltene Zustand $-x$ hat. Ferner wird eine lokal minimale Lösung y durch lokale Suche mit Startlösung x mit der gleichen Wahrscheinlichkeit erreicht, mit der lokale Suche desselben Typs ausgehend von $-x$ in $-y$ terminiert. Insbesondere ist unter jedem n der Erfolg von Antimehrheitsbildung gleich dem Erfolg von Mehrheitsbildung. Die Erfolgsquoten von Antimehrheitsbildung wurden deshalb nicht eigens berechnet.

In Abbildung 3.4 sind die Differenzen der durchschnittlichen f -Werte aus den Mehrheitsläufen (bzw. den Proporz- bzw. Einstimmigkeitsläufen) und den zufälligen Läufen lokaler Suche als „Kurven“ in Abhängigkeit von n skizziert. Die linke Graphik zeigt die Daten unter der probabilistisch wiederholten zyklischen schnellen lokalen Suche, die rechte Graphik unter der deterministischen Variante. Im Mittel werden die f -Werte der lokal minimalen Lösungen von den zufälligen Läufen über die Einstimmigkeitsläufe und die Mehrheitsläufe hin zu den Proporzläufen immer besser. Insbesondere weist die Mehrheitsbildung Werte zwischen denen der Einstimmigkeitsbildung und der Proporzbildung auf. Bei den Kurven der Schrittzahlquotienten (Abbildung 3.5) stellt sich dagegen dieselbe „Rangordnung“ wie beim Problem Max-3-SAT bzw. beim Shiftregister-Problem ein: Proporzläufe machen im Mittel mehr Schritte als Mehrheitsläufe, aber weniger als Einstimmigkeitsläufe. Alle Schrittzahlquotienten sind kleiner als 1. Man beachte, dass Mehrheitsläufe und Antimehrheitsläufe auch bei den f -Wert-Differenzen und den Schrittzahlquotienten im Mittel dieselben Werte erreichen.

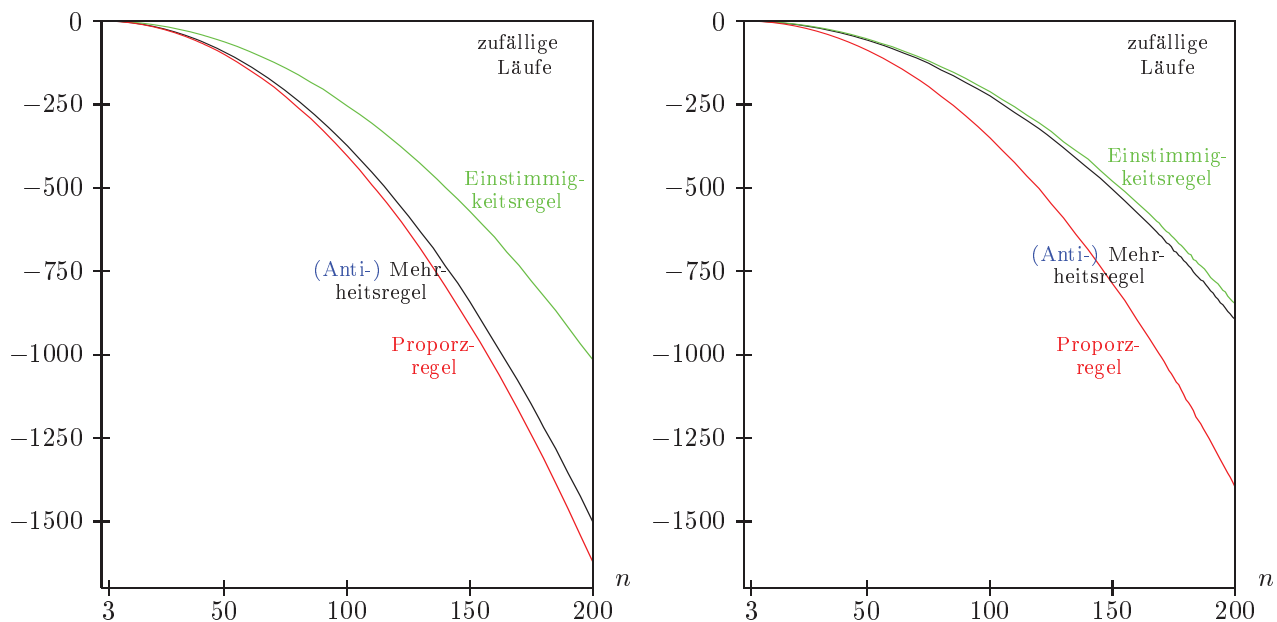


Abbildung 3.4: **(Ising-Spinglas-Problem)** f -Wert-Differenzen von (Anti-) Mehrheitsbildung, Einstimmigkeits- und Proporzbildung in Abhängigkeit von n unter Gruppengröße 3 und probabilistisch (links) bzw. deterministisch (rechts) wiederholter zyklischer schneller lokaler Suche

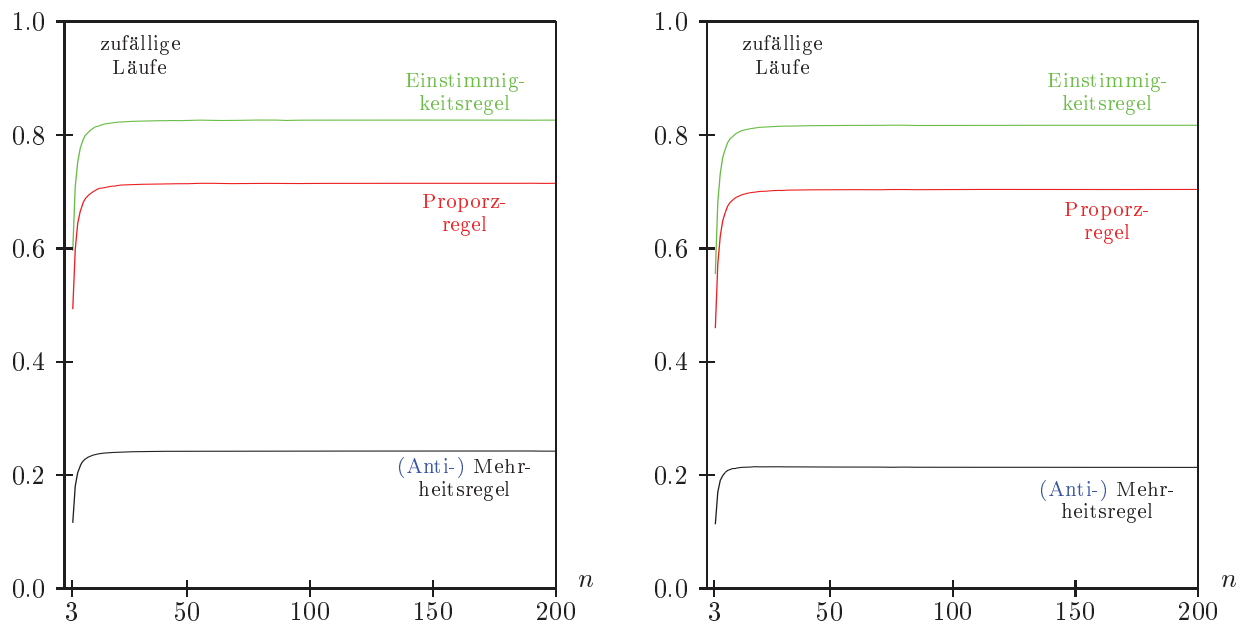


Abbildung 3.5: **(Ising-Spinglas-Problem)** Kurven der Schrittzahlquotienten von (Anti-) Mehrheitsbildung, Einstimmigkeitsbildung und Proporzbildung in Abhängigkeit von n unter Gruppengröße 3 und probabilistisch (links) bzw. deterministisch (rechts) wiederholter zyklischer schneller lokaler Suche

Für das Ising-Spinglas-Problem sind analoge Untersuchungen auch unter zusätzlicher Erlaubnis des Flippens eines Spinzustands durchgeführt worden. Man erinnere sich dazu an

den in Abschnitt 2.2.4 eingeführten Äquivalenzbegriff für Spinzustände sowie die Varianten (V0), (V1) und (V2) der Mehrheitsbildung. Diese Varianten wurden auf die Proporzbildung sowie die Einstimmigkeitsbildung übertragen. Analog zu den Resultaten der Mehrheitsbildung stellte sich jeweils eine Zunahme der Erfolgsquoten von Variante (V0) über (V1) hin zu (V2) ein. Im Vergleich zu den quantitativen Unterschieden der Erfolgsquoten hinsichtlich der verschiedenen Vektorbildungsregeln laut Abbildung 3.3 ist die Zunahme der Erfolgsquoten durch die Varianten (V1) und (V2) jedoch eher klein. Es wird daher von einer detaillierten Darstellung der Daten abgesehen.

Das beobachtete Konvergenzverhalten der behandelten Erfolgskurven wird nun zusammenfassend als Vermutung für den entsprechenden Erfolg formuliert:

VERMUTUNG 3.3

Der Erfolg von Mehrheitsbildung, Einstimmigkeitsbildung und Proporzbildung für

- *Max-3-SAT*
- *das Dynamische Optimierungsproblem*
- *das Shiftregister-Problem (jeweils mit Varianten (V0), (V1), (V2))*
- *das ± 1 -Ising-Spinglas-Problem ohne äußeres Feld (jeweils mit Varianten (V0), (V1), (V2))*

konvergiert unter jeder Gruppengröße $L \geq 2$ und beliebiger in Abschnitt 1.1 genannter Variante wiederholter lokaler Suche für $n \rightarrow \infty$ gegen 1.

Man beachte, dass die Vermutung allgemeiner gehalten ist, als sie durch obige Beobachtungen gefestigt erscheint. Diese Pauschalisierung erhält ihre Rechtfertigung aus dem gesammelten Erfahrungsschatz im Umgang mit den behandelten Regeln.

Am Ende dieses Abschnitts wird der Frage nachgegangen, ob ein Einsatz der Antimehrheitsbildung beim Rucksack-Problem zum Erfolg führt. Wie in Abschnitt 2.2.5 gesehen, fallen die Erfolgsquoten der Mehrheitsbildung für das Rucksack-Problem ab einer bestimmten Anzahl n an Paketen unter die 0.5-Marke. Es erhebt sich die Frage, ob die Erfolgskurve der Antimehrheitsbildung größere Werte als 0.5 annimmt. Das Ergebnis einer entsprechenden Untersuchung ist in Abbildung 3.6 links veranschaulicht. Die Graphik zeigt die Erfolgskurve von Mehrheitsbildung (schwarz, vgl. auch Abbildung 2.24) sowie von Antimehrheitsbildung (blau) unter Gruppengröße 3 und probabilistisch wiederholter zyklischer schneller lokaler Suche. Zur Berechnung der Erfolgsquoten von Antimehrheitsbildung wurden für $n \in \{1, 2, 3, \dots, 200\} \cup \{205, 210, 215, \dots, 350\} \cup \{400, 450, 500\}$ je 100 000 unabhängig und zufällig gewählte Probleminstanzen herangezogen. Anhand des gezeigten Fensters kann das Konvergenzverhalten der Erfolgskurve von Antimehrheitsbildung nicht vollständig geklärt werden. Trotzdem zeigt sich für $n \rightarrow \infty$ ein Wachstum über die 0.50-Marke hinaus. Auch unter den Parametern des Misserfolgstals von Mehrheitsbildung sind die Erfolgsquoten der Antimehrheitsbildung wenigstens teilweise größer als 0.5.

Betrachte nun die durchschnittlichen, mittels Division durch die maximal mögliche Größe eines Pakets (vgl. Abschnitt 2.2.5) normierten Zielfunktionswerte der gefundenen lokal maximalen Lösungen. Der Durchschnittswert der lokal maximalen Lösungen aus den zu-

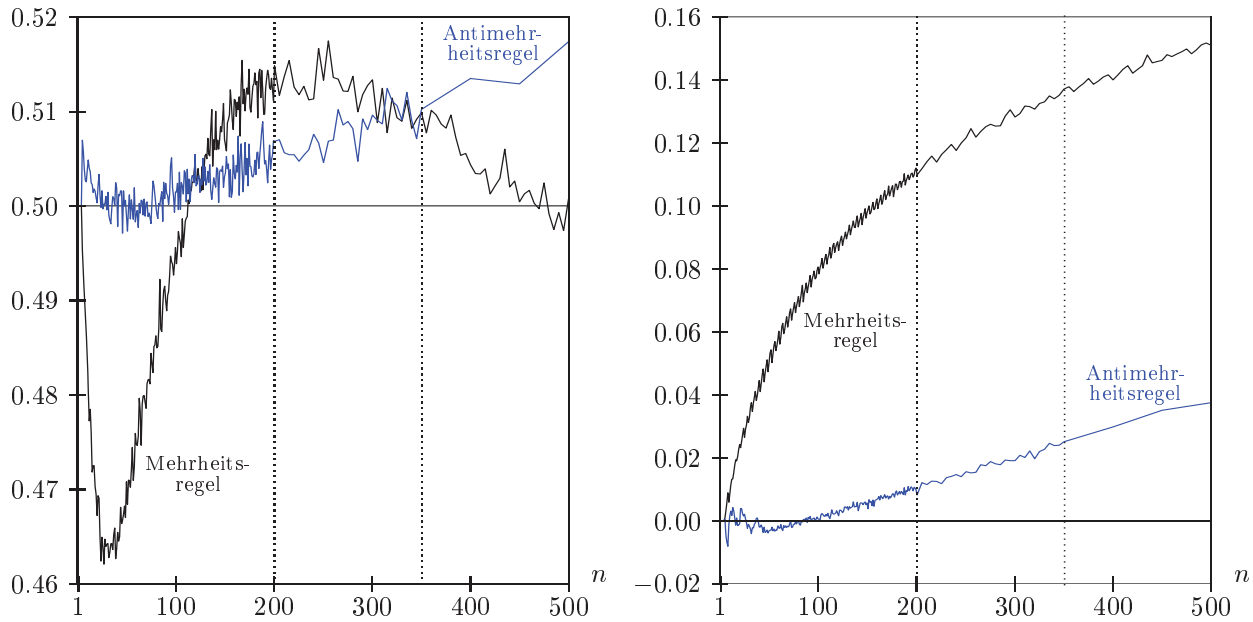


Abbildung 3.6: **(Rucksack-Problem)** Erfolgskurven (links) und normierte f -Wert-Differenzen (rechts) von Mehrheitsbildung und Antimehrheitsbildung in Abhängigkeit von n unter Gruppengröße 3 und probabilistisch wiederholter zyklischer schneller lokaler Suche

fälligen Läufen sei mit \bar{f}_{zf} bezeichnet, aus den Mehrheitsläufen mit \bar{f}_{Mh} und aus den Antimehrheitsläufen mit \bar{f}_{AMh} . In Abbildung 3.6 (rechts) sind die „Kurven“ der Differenzen $\bar{f}_{\text{Mh}} - \bar{f}_{\text{zf}}$ (schwarz) bzw. $\bar{f}_{\text{AMh}} - \bar{f}_{\text{zf}}$ (blau) skizziert. Beide Differenzen sind jeweils ab einem gewissen Problemparameter größer als 0. Unter hinreichend großen Parametern n sind die Erfolgsquoten der Mehrheitsbildung kleiner als 0.5, die Erfolgsquoten der Antimehrheitsbildung dagegen größer als 0.5. Überraschenderweise sind die f -Wert-Differenzen der Mehrheitsbildung für solche n trotzdem größer als diejenigen der Antimehrheitsbildung. Eine mögliche Erklärung für diesen scheinbaren Widerspruch ist folgende:

Die f -Werte der lokal maximalen Lösungen aus den Antimehrheitsläufen streuen um ihren Mittelwert \bar{f}_{AMh} stärker als die f -Werte der lokal maximalen Lösungen aus den zufälligen Läufen um \bar{f}_{zf} . Dieser Unterschied im Streuverhalten reicht dazu aus, dass die Wahrscheinlichkeit, aus drei zufälligen Läufen und einem Antimehrheitslauf einen größeren Maximalwert als aus vier zufälligen Läufen zu erzielen, größer als $\frac{1}{2}$ ist. Auf analoge Weise ist die Streuung der f -Werte der lokal maximalen Lösungen aus den Mehrheitsläufen um ihren Durchschnittswert \bar{f}_{Mh} entsprechend geringer als die der f -Werte aus den zufälligen Läufen um \bar{f}_{zf} . Die Erfolgsquote der Mehrheitsbildung ist dann kleiner als 0.5.

Die Schrittzahlquotienten von Antimehrheitsbildung sind unter jedem Problemparameter n größer als 1. Auf die Darstellung der Kurve der Schrittzahlquotienten wird aus Platzgründen verzichtet. Die „Kurve“ konvergiert mit einem Grenzwert von rund 1.4. Antimehrheitsläufe machen also auch beim Rucksack-Problem im Durchschnitt mehr Schritte als zufällige Läufe.

3.2 Das allgemeine Modell randomisierter Mehrheitsbildungen

Seien $u = (u_1, \dots, u_n)$, $v = (v_1, \dots, v_n)$, $w = (w_1, \dots, w_n) \in \{0, 1\}^n$. Den drei Vektoren u, v, w werde auf folgende Art ein Vektor $x = (x_1, \dots, x_n) \in \{0, 1\}^n$ zugeordnet: Für $j = 0, 1, 2, 3$ sei $p_j \in [0, 1]$. Dann werden die Komponenten x_i ($i = 1, \dots, n$) unabhängig und zufällig mit Wahrscheinlichkeit $p_{u_i+v_i+w_i}$ auf 1 und mit Wahrscheinlichkeit $1 - p_{u_i+v_i+w_i}$ auf 0 gesetzt. p_j gibt also die Wahrscheinlichkeit an, mit der in der i -ten Komponente von x eine 1 auftritt, wenn in den i -ten Komponenten von u, v, w genau j Einsen vorkommen. Tabelle 3.1 enthält die Wahrscheinlichkeiten p_j für die aus dem vorangegangenen Abschnitt bekannten Spezialfälle unter Gruppengröße 3.

p_0	p_1	p_2	p_3	Bildungsregel
0	0	1	1	Mehrheitsregel
0	$\frac{1}{2}$	$\frac{1}{2}$	1	Einstimmigkeitsregel
0	$\frac{1}{3}$	$\frac{2}{3}$	1	Proporzregel
1	1	0	0	Antimehrheitsregel

Tabelle 3.1: Beispiele für randomisierte Mehrheitsbildung

Bei Max- k -SAT, dem Shiftregister-Problem oder dem Ising-Spanglas-Problem ohne äußeres Feld ist es aus Symmetriegründen sinnvoll, $p_3 = 1 - p_0$ und $p_2 = 1 - p_1$ zu fordern. p_0 und p_1 dürfen dagegen beliebig sein.

DEFINITION 3.4

Seien $p_0, p_1 \in [0, 1]$, $p_2 := 1 - p_1$, $p_3 := 1 - p_0$ und $u, v, w \in \{0, 1\}^n$. Die i -te Komponente von u (bzw. v bzw. w) sei mit u_i (bzw. v_i bzw. w_i) bezeichnet. Weiter sei $j_i := u_i + v_i + w_i$ für jedes $i \in \{1, \dots, n\}$. Dann werde $x = (x_1, \dots, x_n)$ gemäß der folgenden (p_0, p_1) -**Regel** gebildet:

$$\begin{aligned} x_i &= 1 && \text{mit Wahrscheinlichkeit } p_{j_i} \\ x_i &= 0 && \text{mit Wahrscheinlichkeit } 1 - p_{j_i} \end{aligned}$$

In diesem Fall heißt x ein (p_0, p_1) -**Vektor zu** u, v, w .

Erfolg, Erfolgsquote und Schrittzahlquotient der (p_0, p_1) -Regel werden analog zu den bekannten Begriffen der Mehrheitsbildung definiert. Im Folgenden werden experimentelle Resultate zu den (p_0, p_1) -Regeln für Max-3-SAT und das Ising-Spanglas-Problem¹ vorgestellt. Bei Max-3-SAT wurden zu einer festen Variablenzahl n stets $\lceil 4.25n \rceil$ Klauseln gebildet. Die Versuche wurden dann mit der probabilistisch wiederholten zyklischen schnellen lokalen Suche durchgeführt, und zwar jeweils für alle $(p_0, p_1) \in \{0.00, 0.05, 0.10, \dots, 1.00\}^2$.

Betrachte zunächst Abbildung 3.7. Diese zeigt die Erfolgsquoten der (p_0, p_1) -Regel für Max-3-SAT unter 100 Variablen (links) bzw. 1000 Variablen (rechts). Für die Graphik

¹Beim Ising-Spanglas-Problem auf dem $(n \times n)$ -Torus bezieht sich die Nomenklatur aus Definition 3.4 auf den mittels der Bijektion φ (siehe S. 27) zu $\{0, 1\}^{n^2}$ transformierten Lösungsraum.

links wurden zu jedem Paar (p_0, p_1) genau 100 000 unabhängige, zufällig gewählte Instanzen gebildet. Die Berechnungen für die rechte Graphik basierten auf 10 000 Instanzen für jedes Paar (p_0, p_1) . Beide Darstellungen haben eins gemeinsam: Die Erfolgsquoten der $(0, 0)$ -Regel (= Mehrheitsregel) sind „beinahe maximal“.² Mit wachsendem p_0 bzw. p_1 fallen ihre Werte relativ schnell ab. Die Erfolgsquoten der $(1, 1)$ -Regel (= Antimehrheitsregel) sind am unteren Ende der Erfolgsskala zu finden.

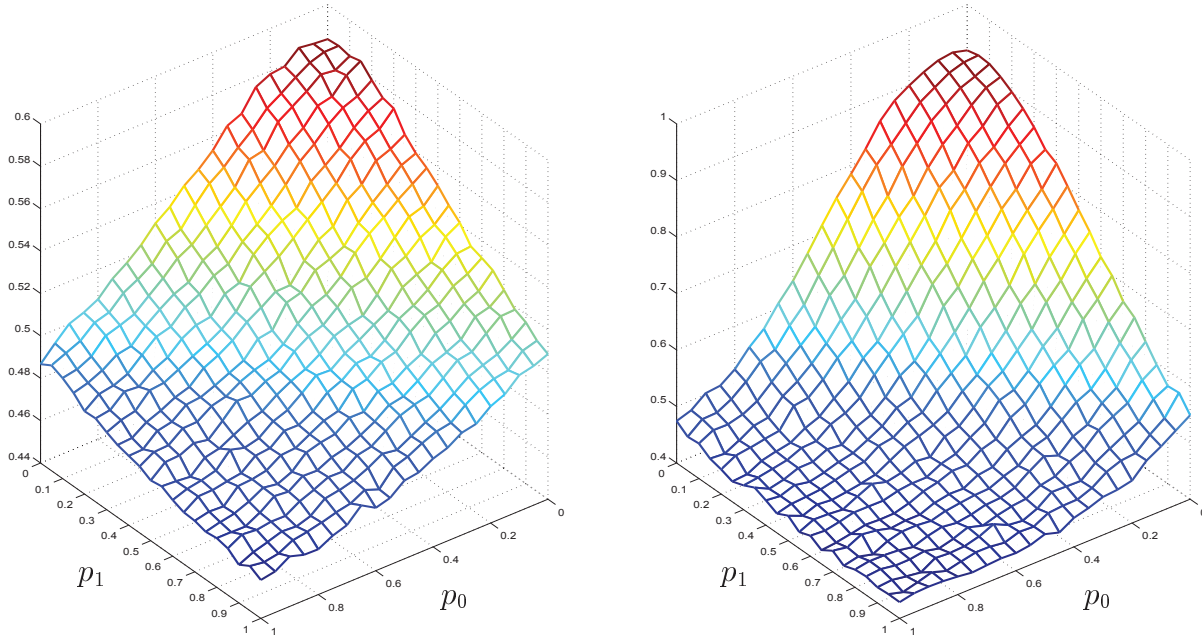


Abbildung 3.7: **(Max-3-SAT)** Erfolgsquoten der (p_0, p_1) -Regel unter $n = 100$ (links) bzw. $n = 1000$ (rechts) und probabilistisch wiederholter zyklischer schneller lokaler Suche

Ein ähnliches Bild ergeben die durchschnittlichen Zielfunktionswerte der lokal maximalen Lösungen aus den Läufen mit den (p_0, p_1) -Vektoren als Startlösung (Abbildung 3.8 links). Die Durchschnittswerte der Mehrheitsläufe sind maximal, die Werte der Antimehrheitsläufe minimal. Da sich die Graphiken unter 100 bzw. 1000 Variablen qualitativ kaum unterscheiden, sind nur die Werte unter 1000 Variablen illustriert. Abbildung 3.8 (rechts) zeigt die Schrittzahlquotienten, ebenfalls für 1000 Variable. Man beachte, dass die $(0, 0)$ -Regel aus optischen Gründen hier an der vorderen Ecke liegt. Mehrheitsläufe machen also minimal viele Schritte, Antimehrheitsläufe maximal viele. Eingebettet in das allgemeine (p_0, p_1) -Modell ist Mehrheitsbildung unter großen Problemparametern hinsichtlich jedes der drei Kriterien optimal.

Es stellt sich nun die Frage, welche Aussagen für die kleinen Problemparameter des Misserfolgstats der Mehrheitsbildung zutreffen. Sei zunächst die Zahl der Variablen 10. Die Erfolgsquoten der (p_0, p_1) -Regel sind in Abbildung 3.9 (links) skizziert. Zu jedem Paar (p_0, p_1) wurden 1 000 000 Instanzen gebildet. Man erkennt, dass sich hier die Antimehrheitsregel als optimal erweist. Die Mehrheitsregel schneidet entsprechend schlecht ab. Mit dem Vorwissen aus Abbildung 3.1 ist diese Beobachtung nicht überraschend. Aus

²Für $p_0 > 0$ oder $p_1 > 0$, aber beide nahe bei 0, sind die Erfolgsquoten geringfügig größer als für $p_0 = 0$ und $p_1 = 0$.

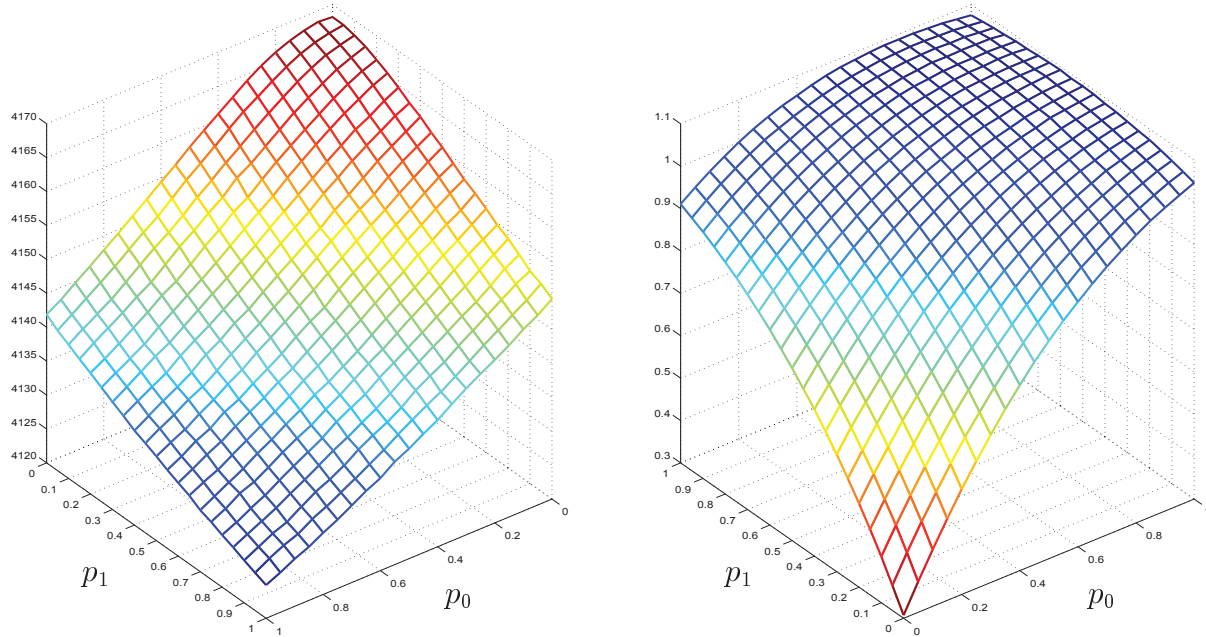


Abbildung 3.8: **(Max-3-SAT)** Durchschnittliche f -Werte (links) bzw. Schrittzahlquotienten (rechts) der (p_0, p_1) -Regel unter $n = 1000$ und probabilistisch wiederholter zyklischer schneller lokaler Suche

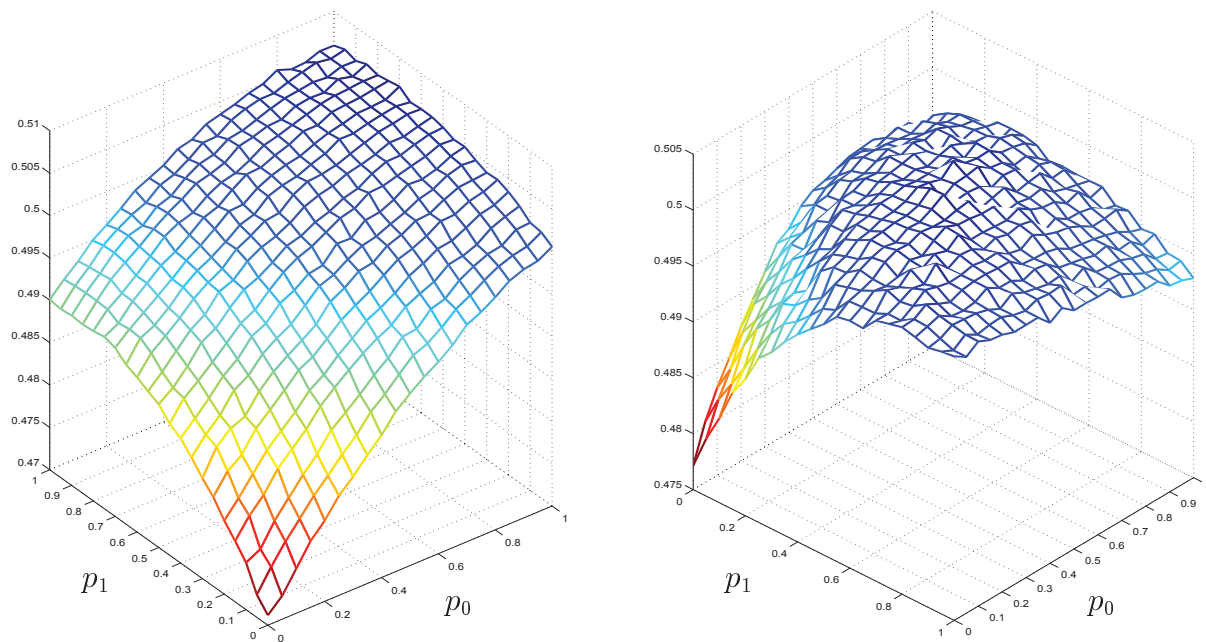


Abbildung 3.9: **(Max-3-SAT)** Erfolgsquoten der (p_0, p_1) -Regel unter $n = 10$ (links) bzw. $n = 17$ (rechts) und probabilistisch wiederholter zyklischer schneller lokaler Suche

Abbildung 3.1 geht aber auch hervor, dass es einen Bereich des Misserfolgstals der Mehrheitsbildung gibt, in dem auch die Erfolgsquoten der Antimehrheitsbildung kleiner als 0.5 sind. Ein Beispiel für einen zugehörigen Problemparameter ist Variablenzahl 17. Es wäre wünschenswert, wenn auch bei 17 Variablen ein Paar (p_0, p_1) existierte, für das die Erfolgsquote größer als 0.5 ist. Abbildung 3.9 (rechts) macht diese Hoffnung jedoch zunichte:

Aus der Graphik geht hervor, dass die Mehrheitsregel die schlechtesten Erfolgsquoten besitzt. Die $(\frac{1}{2}, \frac{1}{2})$ -Regel erreicht die größten Werte. Läufe lokaler Suche mit Startvektoren, welche nach der $(\frac{1}{2}, \frac{1}{2})$ -Regel gebildet werden, sind zufällige Läufe. Insbesondere hat die $(\frac{1}{2}, \frac{1}{2})$ -Regel einen Erfolg von 0.5.

Zur Ergänzung sind in den Abbildungen 3.10 bzw. 3.11 die durchschnittlichen Zielfunk-

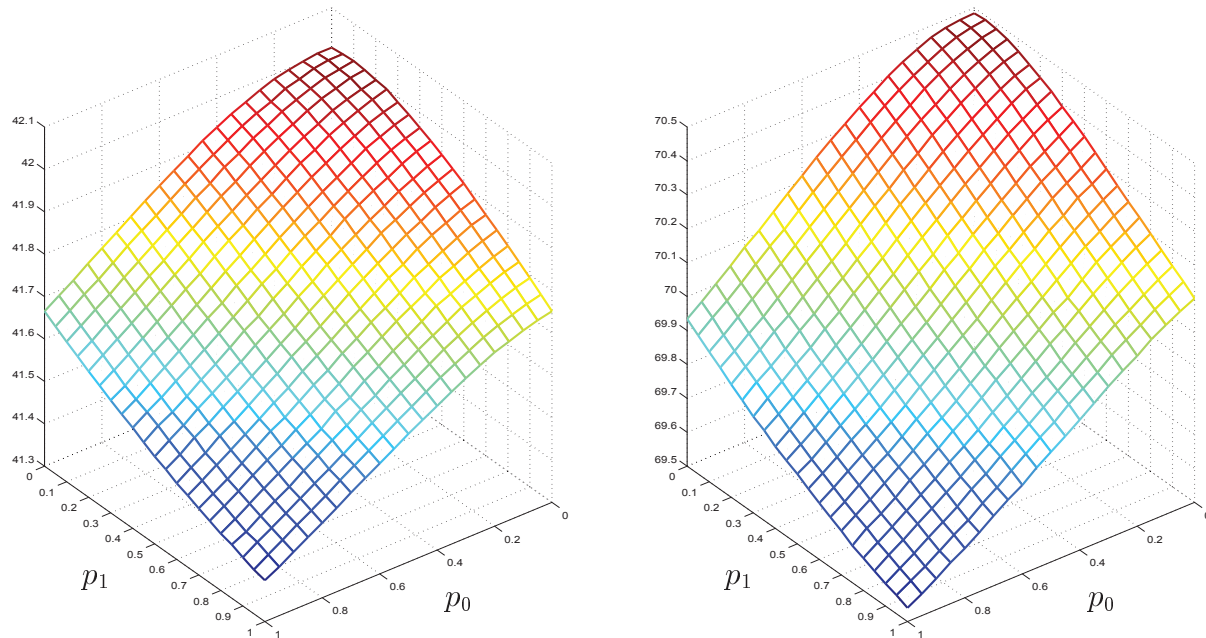


Abbildung 3.10: **(Max-3-SAT)** Durchschnittliche f -Werte der (p_0, p_1) -Regel unter $n = 10$ (links) bzw. $n = 17$ (rechts) und probabilistisch wiederholter zyklischer schneller lokaler Suche

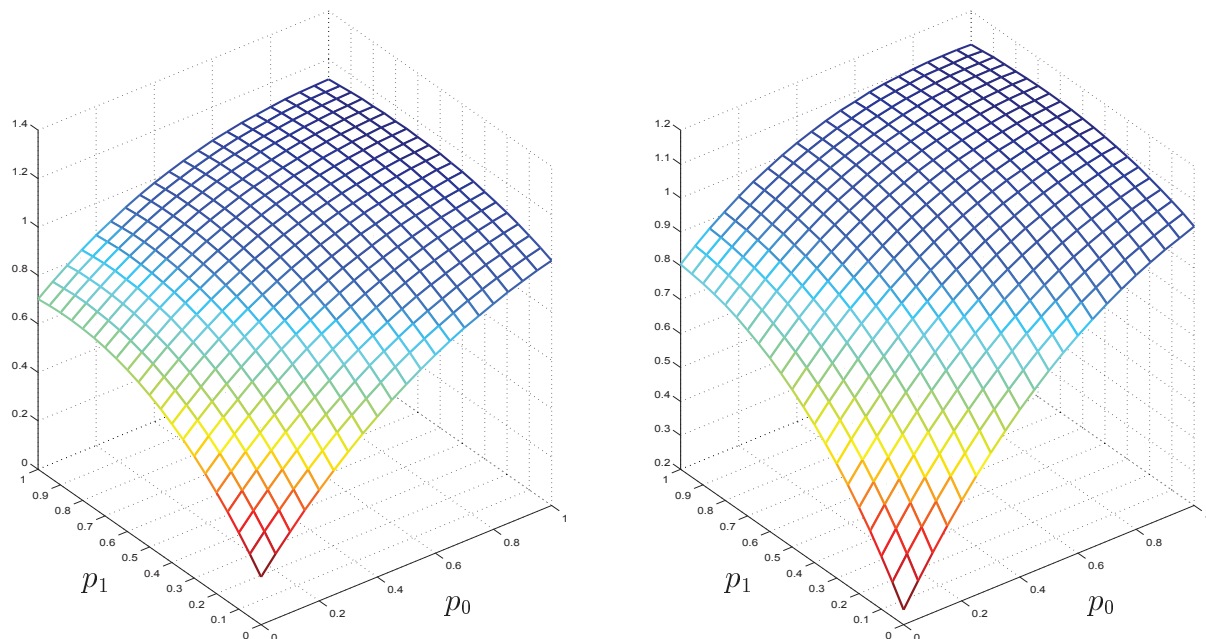


Abbildung 3.11: **(Max-3-SAT)** Schrittzahlquotienten der (p_0, p_1) -Regel unter $n = 10$ (links) bzw. $n = 17$ (rechts) und probabilistisch wiederholter zyklischer schneller lokaler Suche

tionswerte der lokal maximalen Lösungen bzw. die Schrittzahlquotienten dargestellt. Aus den Abbildungen wird ersichtlich, dass die Mehrheitsregel auch unter den Parametern ihres Misserfolgstals in beiderlei Hinsicht optimal ist.

Im Folgenden wird das ± 1 -Ising-Spinglas-Problem auf dem $(n \times n)$ -Torus ohne äußeres Feld behandelt. Die Zielfunktion f jeder Instanz ist im folgenden Sinne symmetrisch: Ist x ein Zustand der n^2 Spins, d.h. eine Belegung mit Elementen aus $\{-1, +1\}$, so gilt $f(-x) = f(x)$. Insbesondere gilt für alle Zustände x und alle lokal minimalen Zustände y : Lokale Suche mit Startzustand x terminiert in y mit derselben Wahrscheinlichkeit, wie lokale Suche der gleichen Variante, beginnend in $-x$, in $-y$ terminiert. Für alle Paare $(p_0, p_1) \in [0, 1]^2$ ist der Erfolg der (p_0, p_1) -Regel somit gleich dem Erfolg der $(1-p_0, 1-p_1)$ -Regel. Analog spielt es für den f -Wert eines lokal minimalen Spinzustands aus lokaler Suche bzw. die erwarteten Schrittzahlquotienten keine Rolle, ob der Startzustand gemäß der (p_0, p_1) - oder der $(1-p_0, 1-p_1)$ -Regel gebildet wird.

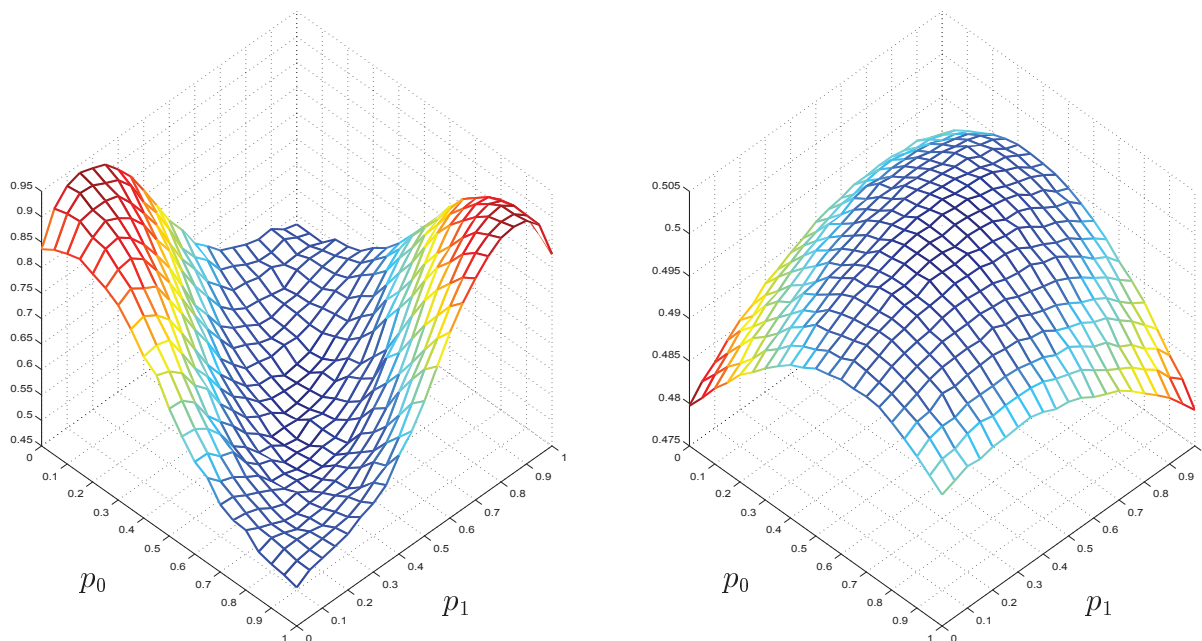


Abbildung 3.12: **(Ising-Spinglas-Problem)** Erfolgsquoten der (p_0, p_1) -Regel unter $n = 50$ (links) bzw. $n = 3$ (rechts) und probabilistisch wiederholter zyklischer schneller lokaler Suche

Die Experimente zu den (p_0, p_1) -Regeln wurden für $n = 50$ und $n = 3$ durchgeführt. Man beachte, dass $n = 3$ dem Misserfolgstal von Mehrheitsbildung angehört. Die Erfolgsquote von Mehrheitsbildung unter $n = 50$ ist dagegen größer als 0.83.³ Für $n = 50$ wurden 10 000 Probleminstanzen pro Paar (p_0, p_1) erzeugt, für $n = 3$ jeweils 1 000 000. Die erhaltenen Erfolgsquoten der (p_0, p_1) -Regeln sind in Abbildung 3.12 veranschaulicht. Sie zeigen, dass Paare (p_0, p_1) in einer Umgebung von $(\frac{1}{2}, \frac{1}{2})$ bei $n = 50$ die schlechtesten Erfolgsquoten liefern, bei $n = 3$ dagegen die besten. Die $(0, 0)$ -Regel (Mehrheitsregel) hat umgekehrt bei $n = 50$ eher gute, bei $n = 3$ schlechte Werte. Bei Parametern des Misserfolgstals von Mehrheitsbildung sind also zufällige Läufe lokaler Suche ratsam. Bei $n = 50$ ergaben die $(0.00, 0.15)$ - sowie die $(1.00, 0.85)$ -Regel die größten Erfolgsquoten unter allen getesteten Werten. Hier sind die Erfolgsquoten größer als 0.93. Diese beiden Regeln sind auch hin-

³vgl. Abbildung 2.20 links bzw. 3.3 links

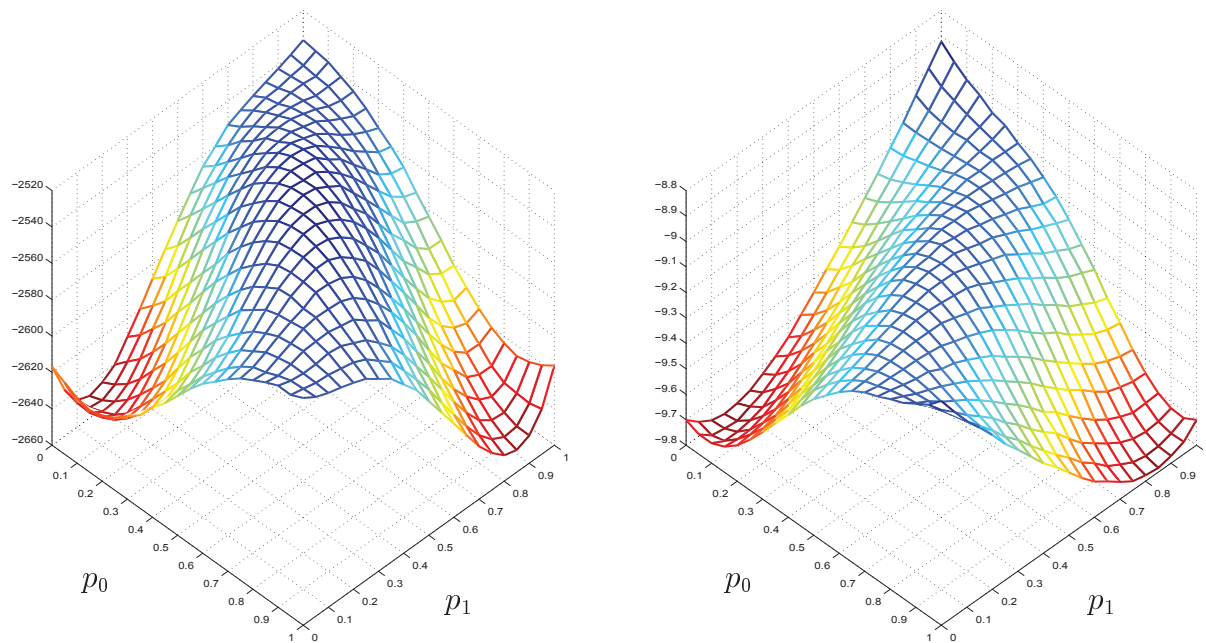


Abbildung 3.13: **(Ising-Spinal-Problem)** Durchschnittliche f -Werte der (p_0, p_1) -Regel unter $n = 50$ (links) bzw. $n = 3$ (rechts) und probabilistisch wiederholter zyklischer schneller lokaler Suche

sichtlich der durchschnittlichen f -Werte optimal (Abbildung 3.13), und zwar sowohl für $n = 50$ als auch für $n = 3$. Bei den Schrittzahlquotienten liefert die Mehrheits- bzw. Antimehrheitsregel die kleinsten Werte (Abbildung 3.14).

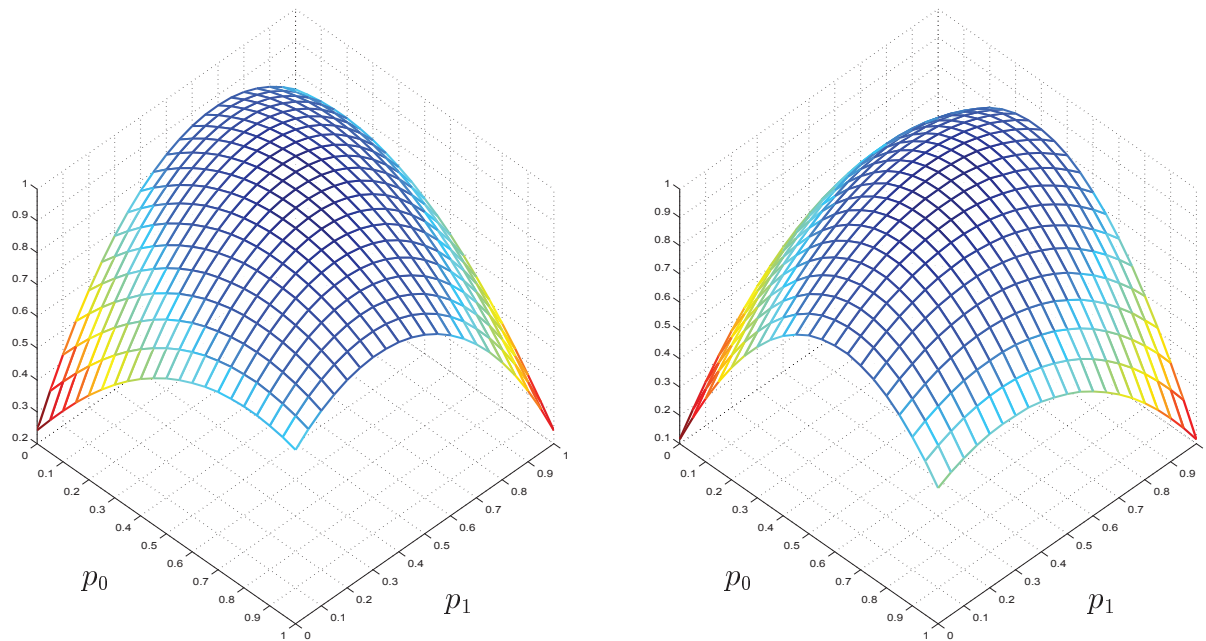


Abbildung 3.14: **(Ising-Spinal-Problem)** Schrittzahlquotienten der (p_0, p_1) -Regel unter $n = 50$ (links) bzw. $n = 3$ (rechts) und probabilistisch wiederholter zyklischer schneller lokaler Suche

Kapitel IV

Mehrstufige Mehrheitsbildung

In Kapitel II wurde für verschiedene Optimierungsprobleme beobachtet, dass Mehrheitsläufe im Durchschnitt zu besseren Lösungen führen als zufällige Läufe lokaler Suche. Als Startlösung eines Mehrheitslaufs wurde dabei ein Mehrheitsvektor zu lokal maximalen Lösungen aus zufälligen Läufen gewählt. Solche Läufe seien im Folgenden als **Mehrheitsläufe erster Stufe** bezeichnet. Es wird nun untersucht, ob die Güte der Zielfunktionswerte noch weiter gesteigert werden kann, wenn zur Mehrheitsbildung lokal maximale Lösungen aus Mehrheitsläufen statt aus zufälligen Läufen herangezogen werden. Zu ersten Versuchszwecken wird auf das aus Abschnitt 2.1.1 bekannte Max-SAT-Benchmark „Hanoi 5“ zurückgegriffen.

4.1 Mehrstufige Mehrheitsbildung bei „Hanoi 5“

Zur Untersuchung einstufiger Mehrheitsbildung am Beispiel „Hanoi 5“ in Abschnitt 2.1.1 wurden 2000 Läufe einer fest gewählten Variante lokaler Suche gemacht. Die Läufe unterteilten sich in 500 Blöcke zu je drei zufälligen Läufen und einem Mehrheitslauf. Für die folgenden Experimente bestehe ein Block aus insgesamt 13 Läufen, nämlich aus neun zufälligen Läufen und vier Mehrheitsläufen. Die lokal maximalen Lösungen der zufälligen Läufe seien mit x^1, x^2, \dots, x^9 bezeichnet. Dann wird als Startlösung von Lauf 10 der Mehrheitsvektor zu x^1, x^2, x^3 verwendet. Analog startet Lauf 11 mit dem Mehrheitsvektor zu x^4, x^5, x^6 und Lauf 12 mit dem Mehrheitsvektor zu x^7, x^8, x^9 . Die Läufe 10, 11 und 12 sind also Mehrheitsläufe erster Stufe. Seien x^{10}, x^{11}, x^{12} die entsprechenden lokal maximalen Lösungen. Dann wird als Startlösung von Lauf 13 schließlich der Mehrheitsvektor zu x^{10}, x^{11}, x^{12} herangezogen. Lauf 13 ist ein **Mehrheitslauf zweiter Stufe**. Um die Zielfunktionswerte der Läufe mit den Ergebnissen in Abschnitt 2.1.1 vergleichbar zu machen, werden auch bei diesem zweistufigen Verfahren genau 2000 Läufe zugelassen. Diese unterteilen sich in 153 Blöcke zu je 13 Läufen und einen verkürzten Block mit 11 Läufen. Im letzten Block fehlt ein Mehrheitslauf erster Stufe sowie der Mehrheitslauf zweiter Stufe.

Es wurden dieselben Varianten wiederholter lokaler Suche wie in Abschnitt 2.1.1 verwendet, nämlich deterministisch wiederholte zyklische schnelle lokale Suche, probabilistisch wiederholte zyklische schnelle lokale Suche und wiederholte volle lokale Suche, jeweils ohne und mit Sidesteps. In den Abbildungen 4.1 bis 4.6 sind die Häufigkeitsverteilungen der gefundenen lokal maximalen Lösungen dargestellt. Bei jeder Variante wiederholter lokaler Suche waren alle 2000 gefundenen lokal maximalen Lösungen verschieden.

Die schwarzen Säulen der Diagramme zeigen die Häufigkeiten der f -Werte aus den zufälligen Läufen. Die roten Säulen beziehen sich auf die Mehrheitsläufe erster Stufe. Die blauen Säulen kennzeichnen die Mehrheitsläufe zweiter Stufe. In den sechs Balkendia-

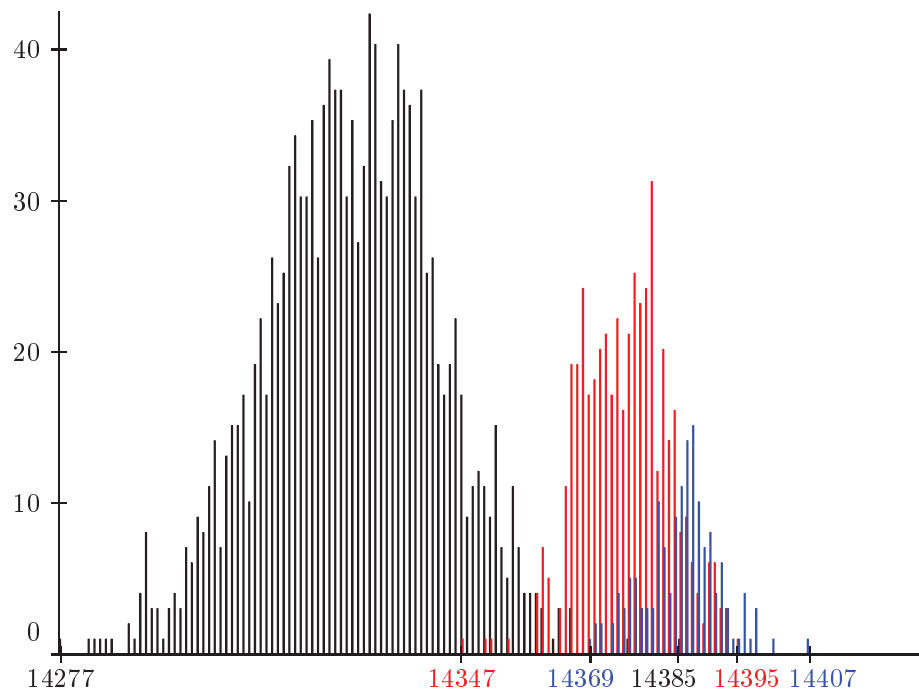


Abbildung 4.1: Häufigkeitsverteilung der gefundenen lokalen Maxima bei zweistufiger Mehrheitsbildung und deterministisch wiederholter zyklischer schneller lokaler Suche (ohne Sidesteps)

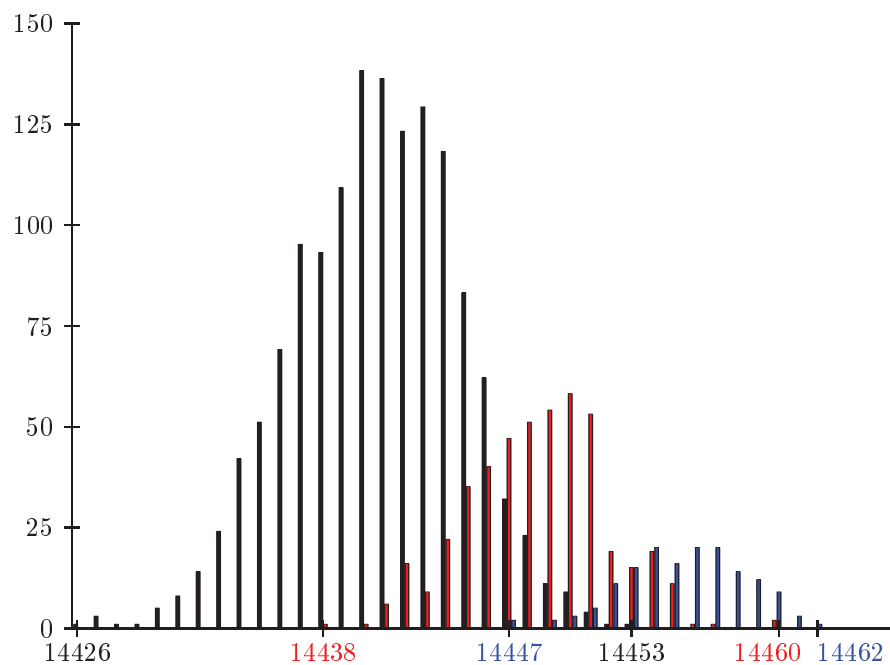


Abbildung 4.2: Häufigkeitsverteilung der gefundenen lokalen Maxima bei zweistufiger Mehrheitsbildung und deterministisch wiederholter zyklischer schneller lokaler Suche (mit Sidesteps)

grammen sind ferner für jeden Lauftyp der beste und schlechteste gefundene f -Wert in entsprechender Farbe markiert. Das wichtigste Zahlenmaterial der Versuchsausgänge ist Tabelle 4.1 zu entnehmen. Die Bedeutung der verwendeten Bezeichnungen ist wie in Tabelle 2.1, Seite 34. Die einzige Abweichung von den dortigen Schreibweisen besteht in

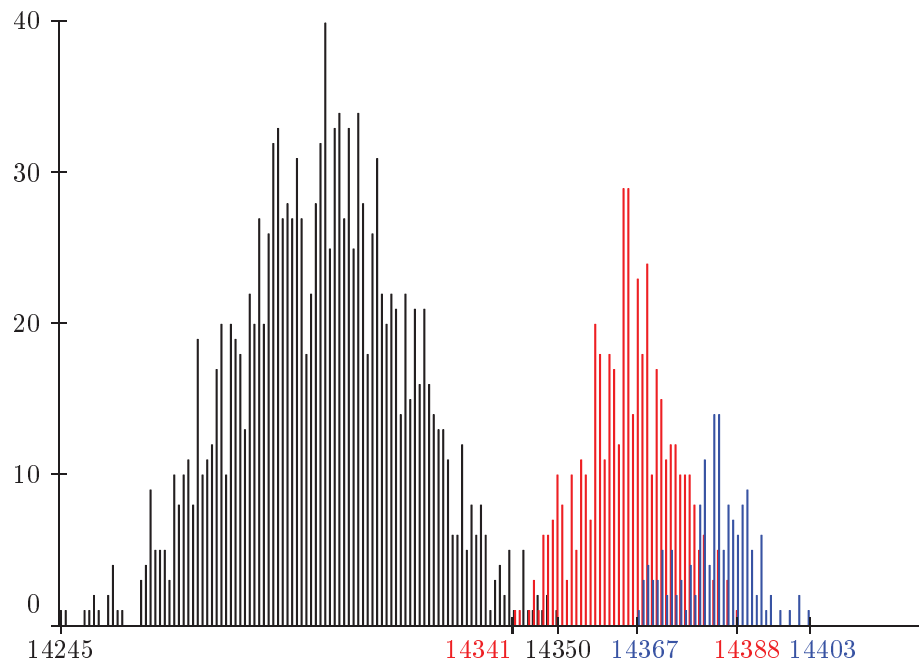


Abbildung 4.3: Häufigkeitsverteilung der gefundenen lokalen Maxima bei zweistufiger Mehrheitsbildung und probabilistisch wiederholter zyklischer schneller lokaler Suche (ohne Sidesteps)

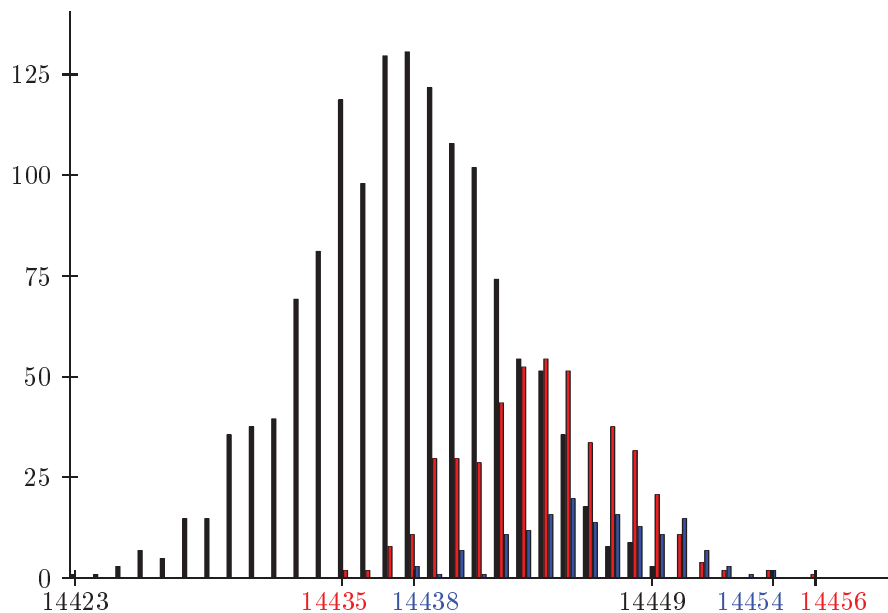


Abbildung 4.4: Häufigkeitsverteilung der gefundenen lokalen Maxima bei zweistufiger Mehrheitsbildung und probabilistisch wiederholter zyklischer schneller lokaler Suche (mit Sidesteps)

der Unterscheidung zwischen Mehrheitsläufen erster Stufe („Mh1“) und Mehrheitsläufen zweiter Stufe („Mh2“). Zusammenfassend lassen sich folgende Beobachtungen machen:

1. Mehrheitsläufe erster Stufe erzielen im Durchschnitt bessere Zielfunktionswerte als zufällige Läufe. Mehrheitsläufe zweiter Stufe führen im Mittel zu besseren Werten als Mehrheitsläufe erster Stufe. Die besten gefundenen f -Werte stammen typischerweise aus Mehrheitsläufen zweiter Stufe.

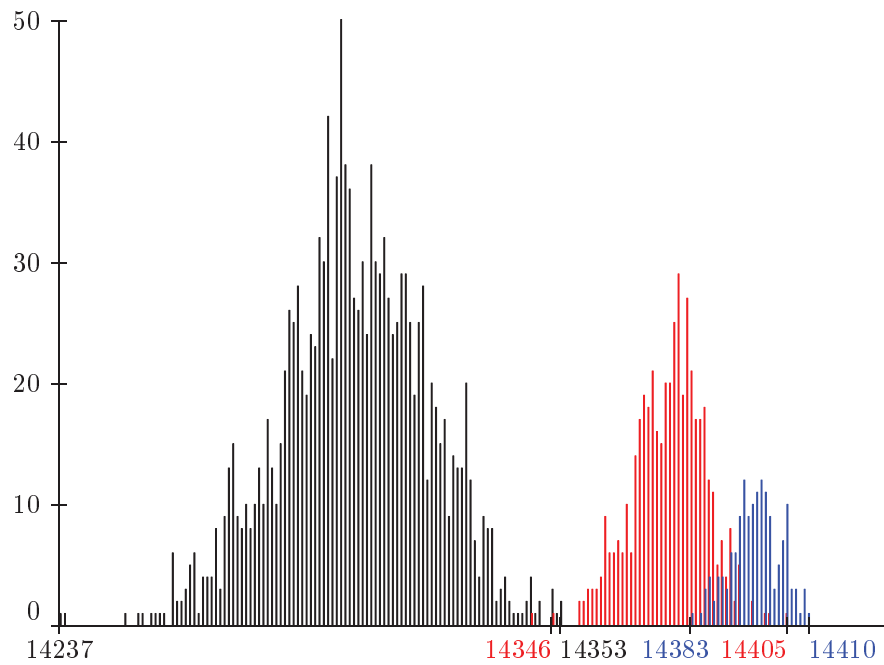


Abbildung 4.5: Häufigkeitsverteilung der gefundenen lokalen Maxima bei zwei-stufiger Mehrheitsbildung und wiederholter voller lokaler Suche (ohne Sidesteps)

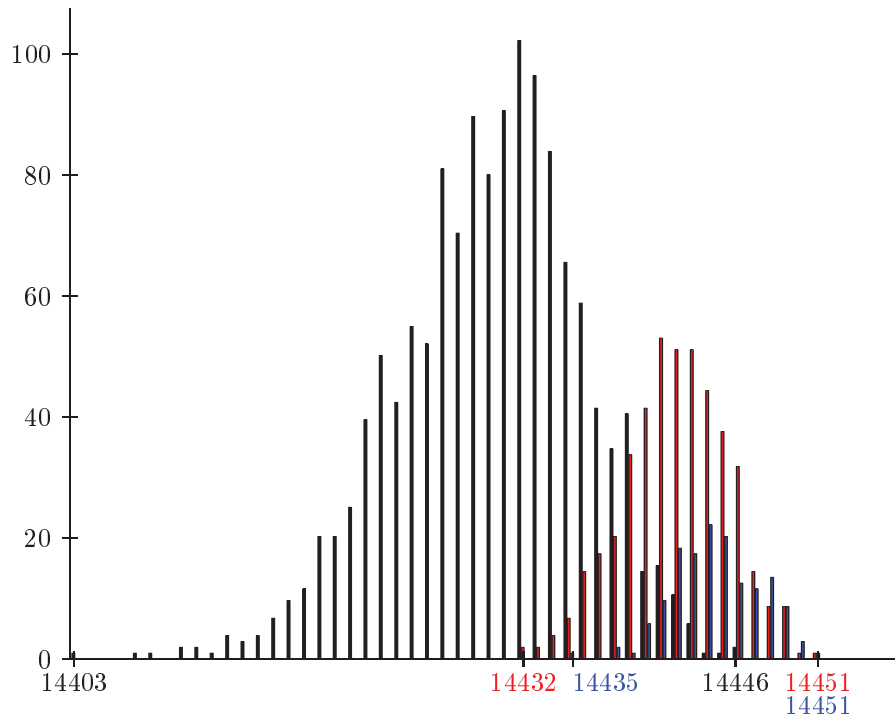


Abbildung 4.6: Häufigkeitsverteilung der gefundenen lokalen Maxima bei zwei-stufiger Mehrheitsbildung und wiederholter voller lokaler Suche (mit Sidesteps)

2. Mehrheitsläufe machen im Durchschnitt weniger Schritte als zufällige Läufe. Bei den Varianten lokaler Suche ohne Sidesteps nimmt die durchschnittliche Schrittzahl von den Mehrheitsläufen erster Stufe zu den Mehrheitsläufen zweiter Stufe weiter ab.
3. Mehrheitsläufe stellen im Durchschnitt weniger Fragen als zufällige Läufe. Bei den

	\max_{zf}	\max_{Mh1}	\max_{Mh2}	\overline{f}_{zf}	\overline{f}_{Mh1}	\overline{f}_{Mh2}
dZS	14385	14395	14407	14327.9	14375.3	14385.0
dZSs	14453	14460	14462	14440.6	14448.4	14455.5
pZS	14350	14388	14403	14300.3	14365.0	14383.2
pZSs	14449	14456	14454	14437.6	14443.9	14446.0
V	14353	14405	14410	14304.9	14377.7	14397.6
Vs	14446	14451	14451	14429.9	14441.9	14444.5

	\overline{S}_{zf}	\overline{S}_{Mh1}	\overline{S}_{Mh2}	\overline{F}_{zf}	\overline{F}_{Mh1}	\overline{F}_{Mh2}
dZS	512.6	75.6	53.3	12710.7	6341.6	5556.7
dZSs	4341.6	3452.1	3543.6	289949.5	295185.7	348882.8
pZS	504.8	84.2	64.2	10338.3	5918.7	4950.1
pZSs	3994.6	3202.3	3212.5	271667.2	262772.8	269698.1
V	419.0	75.5	56.6	811112.0	147769.7	111177.6
Vs	4281.2	3486.7	3193.0	8269008.7	6734770.9	6167525.7

Tabelle 4.1: Ergebnisse zweistufiger Mehrheitsbildung bei verschiedenen Varianten lokaler Suche bei „Hanoi 5“

Varianten lokaler Suche ohne Sidesteps nimmt die durchschnittliche Fragenzahl von den Mehrheitsläufen erster Stufe zu den Mehrheitsläufen zweiter Stufe weiter ab.

Bei Erlaubnis von Sidesteps lässt sich keine generelle Abnahme der durchschnittlichen Schrittzahlen von den Mehrheitsläufen erster Stufe hin zu den Mehrheitsläufen zweiter Stufe feststellen. Bei *einstufiger* Mehrheitsbildung und probabilistisch wiederholter zyklischer schneller lokaler Suche trat in den Mehrheitsläufen (erster Stufe) jedoch eine durchschnittliche Schrittzahl von 3241.3 statt 3202.3 auf (vgl. Tabelle 2.1, Seite 34). Die höheren Schrittzahlen der Mehrheitsläufe zweiter Stufe gegenüber denen erster Stufe in Tabelle 4.1 sind vermutlich in den zu geringen Anzahlen der Mehrheitsläufe beider Stufen und einer starken Streuung der Schrittzahlen um ihre Mittelwerte begründet.

4.2 Wiederholte Mehrheitsbildung als Genetischer Algorithmus

Die experimentellen Ergebnisse zu „Hanoi 5“ aus Abschnitt 4.1 sprechen für eine zwei- statt einstufige Anwendung der Mehrheitsbildung. Auf beliebige Stufenzahl $S \geq 1$ und Gruppengröße $L \geq 2$ verallgemeinert, erhält man folgendes Verfahren:

In Stufe $t \in \{0, 1, \dots, S\}$ werden L^{S-t} Läufe lokaler Suche gemacht. In Stufe 0 sind alle Läufe zufällig. Die Läufe in Stufe $t > 0$ sind Mehrheitsläufe, deren Startvektoren durch Mehrheitsbildung aus den lokal maximalen Lösungen von Stufe $t - 1$ gebildet werden. Dazu werden jeweils die lokal maximalen Lösungen aus L aufeinanderfolgenden Läufen von Stufe $t - 1$ herangezogen. Bis die lokal maximale Lösung des Mehrheitslaufs aus

Stufe S vorliegt, sind insgesamt

$$\sum_{t=0}^S L^{S-t} = \frac{L^{S+1} - 1}{L - 1}$$

Läufe lokaler Suche nötig. Insbesondere wird ein Großteil der Rechenzeit für die zufälligen Läufe in Stufe 0 verbraucht.

Die Idee ist nun, für eine konstante Anzahl ϱ an Läufen in jeder Stufe zu sorgen. Dazu wird nach jeder Stufe t ein *Pool* der ϱ lokal maximalen Lösungen $x^0, x^1, \dots, x^{\varrho-1}$ eingerichtet, aus denen auf geeignete Weise ϱ Startvektoren für die Läufe in Stufe $t+1$ gebildet werden. Natürlich muss dazu jede Lösung x^ℓ , $\ell = 0, 1, \dots, \varrho-1$, mehrfach herangezogen werden. Die Frage ist also, wie man aus ϱ Lösungen auf geeignete Weise eine Auswahl von ϱ Gruppen zu je L Lösungen trifft. Sinnvoll sind folgende Forderungen an die ϱ Gruppen:

- (i) Die Gruppen sollen paarweise verschieden sein.
- (ii) Jede Gruppe soll Lösungen aus L verschiedenen Läufen enthalten.
- (iii) Jede der ϱ Lösungen soll in gleich vielen Gruppen vorkommen.
- (iv) Falls möglich, sollen je zwei Gruppen Lösungen aus höchstens einem Lauf gemeinsam haben.
- (v) Die Bildung der ϱ Gruppen soll praktikabel sein. Es soll ein allgemeines, einfaches Konstruktionsschema existieren.

Unter bestimmten Bedingungen an ϱ lassen sich die Forderungen erfüllen:

BEMERKUNG 4.1

Sei $L \geq 2$, und sei $\varrho = k \cdot L$ mit $k \geq L$. Für $v \in \{0, \dots, L-1\}$ und $s \in \{0, \dots, k-1\}$ sei $G_{v,s} \subseteq \{0, 1, \dots, \varrho-1\}$ definiert durch

$$G_{v,s} := \{ik + (s + iv) \bmod k \mid i = 0, \dots, L-1\}.$$

Dann erfüllen die Mengen $G_{v,s}$ folgende Eigenschaften:

- (i) Für alle v, v', s, s' mit $(v, s) \neq (v', s')$ gilt $G_{v,s} \neq G_{v',s'}$.
- (ii) Für alle v, s gilt $\#G_{v,s} = L$.
- (iii) Für alle $\ell \in \{0, 1, \dots, \varrho-1\}$ gilt $\#\{(v, s) \mid \ell \in G_{v,s}\} = L$.
- (iv) Für alle v und alle s, s' mit $s \neq s'$ gilt $\#(G_{v,s} \cap G_{v,s'}) = 0$.
Falls k eine Primzahl oder $k > (L-1)^2$ ist, gilt $\#(G_{v,s} \cap G_{v',s'}) \leq 1$ für alle v, v' mit $v \neq v'$ und alle s, s' .

BEWEIS.

- (i) Seien $v, v' \in \{0, \dots, L-1\}$ und $s, s' \in \{0, \dots, k-1\}$ mit $(v, s) \neq (v', s')$.
FALL 1: $s \neq s'$. Mit $i = 0$ zeigt sich $s \in G_{v,s}$. Andererseits ist aber $s \notin G_{v',s'}$, denn aus $s = ik + (s' + iv') \bmod k \in G_{v',s'}$ für ein i würde wegen $s < k$ sofort $i = 0$ und somit $s = s'$ folgen.

FALL 2: $s = s'$. Dann ist $v \neq v'$. Mit $i = 1$ zeigt sich $k + (s + v) \bmod k \in G_{v,s}$. Gälte $k + (s + v) \bmod k \in G_{v',s'}$, also

$$k + (s + v) \bmod k = ik + (s + iv) \bmod k$$

für ein i , so würde $i = 1$, also $(s + v) \bmod k = (s + v') \bmod k$ und somit $v = v'$ folgen.

- (ii) $\#G_{v,s} \leq L$ ist klar. Zu zeigen bleibt $\#G_{v,s} \geq L$. Seien dazu $i, i' \in \{0, \dots, L-1\}$ mit $ik + (s + iv) \bmod k = i'k + (s + i'v) \bmod k$, d.h. $(i - i')k + ((i - i')v) \bmod k = 0$. Dann folgt $i - i' = 0$, d.h. $i = i'$.
- (iv) BEH. 1: $\#(G_{v,s} \cap G_{v',s'}) = 0$, falls $v \in \{0, \dots, L-1\}$ und $s, s' \in \{0, \dots, k-1\}$ mit $s \neq s'$.
BEWEIS. Gäbe es i, i' mit $ik + (s + iv) \bmod k = i'k + (s' + i'v) \bmod k$, d.h.

$$(i - i')k + ((s - s') + (i - i')v) \bmod k = 0,$$

so würde $i - i' = 0$, d.h. $i = i'$, und somit $(s - s') \bmod k = 0$ folgen. Da $s, s' < k$, erhielte man $s - s' = 0$, also $s = s'$ im Widerspruch zur Voraussetzung.

Sei nun k prim oder $k > (L-1)^2$.

BEH. 2: $\#(G_{v,s} \cap G_{v',s'}) \leq 1$ für alle v, v' mit $v \neq v'$ und alle s, s'

BEWEIS. Seien $a, b \in G_{v,s} \cap G_{v',s'}$, etwa

$$\begin{aligned} a &= ik + (s + iv) \bmod k = i'k + (s' + i'v') \bmod k \\ b &= jk + (s + jv) \bmod k = j'k + (s' + j'v') \bmod k \end{aligned}$$

mit $i, i', j, j' \in \{0, \dots, L-1\}$. Zu zeigen ist $a = b$. Aus

$$\begin{aligned} (i - i')k + ((s - s') + iv - i'v') \bmod k &= 0 \\ (j - j')k + ((s - s') + jv - j'v') \bmod k &= 0 \end{aligned}$$

folgt $i = i'$ sowie $j = j'$ und deshalb

$$\begin{aligned} (s - s') + i(v - v') \bmod k &= 0, \\ (s - s') + j(v - v') \bmod k &= 0. \end{aligned}$$

Subtraktion der Gleichungen ergibt

$$((i - j)(v - v')) \bmod k = 0.$$

Zu zeigen bleibt $i = j$. Falls k eine Primzahl ist, ist $\mathbb{Z}/k\mathbb{Z}$ ein Körper und damit insbesondere nullteilerfrei. Aus $(i - j)(v - v') = 0$ in $\mathbb{Z}/k\mathbb{Z}$ folgt wegen $v - v' \neq 0$ deshalb $i - j = 0$. Gelte nun $k > (L-1)^2$. Sei $z \in \mathbb{Z}$ mit

$$|i - j| \cdot |v - v'| = z \cdot k.$$

Wegen $|i - j| \cdot |v - v'| \leq (L-1)^2$ und $k > (L-1)^2$ ergibt sich $z = 0$. Da $v - v' \neq 0$, folgt $i - j = 0$.

- (iii) Seien $\ell \in \{0, \dots, p-1\}$ und $v \in \{0, \dots, L-1\}$. Dann ist zu zeigen, dass genau ein $s \in \{0, \dots, k-1\}$ mit $\ell \in G_{v,s}$ existiert. Seien $i \in \{0, \dots, L-1\}$ und $r \in \{0, \dots, k-1\}$ mit $\ell = ik + r$. Setze $s := (r - iv) \bmod k$. Dann gilt $ik + (s + iv) \bmod k = ik + r = \ell$. Damit ist die Existenz von s nachgewiesen. Die Eindeutigkeit von s gilt nach (iv). \square

Das Bildungsgesetz der ϱ Mengen $G_{v,s}$ ist einfach zu programmieren und anzuwenden. Es wird insbesondere für die nachfolgend beschriebenen Untersuchungen herangezogen.

Falls ϱ nicht die Voraussetzung $\varrho = k \cdot L$ mit $k \geq L$ erfüllt, wird $\varrho' := \max \{L^2, \lceil \frac{\varrho}{L} \rceil \cdot L\}$ gesetzt. Dann werden zu den ϱ bestehenden Lösungen $\varrho' - \varrho$ Kopien hinzugefügt, d.h. der Pool wird auf Größe ϱ' „aufgeblasen“. Sei dazu $\varrho' = k \cdot \varrho + r$ mit $k, r \geq 0$ und $r < \varrho$. Dann werden zunächst k Kopien des gesamten ursprünglichen Pools gebildet. Danach werden Lösungen aus r verschiedenen Läufen zufällig gleichverteilt gewählt. Die erhaltenen ϱ' Lösungen bilden den neuen größeren Pool. Auf diesen wird das Bildungsgesetz gemäß Bemerkung 4.1 angewendet. Von den ϱ' Gruppen werden schließlich die gemäß der lexikographischen Anordnung der Paare (v, s) ersten ϱ Gruppen übernommen. Mit dieser Zusatzregelung können beliebige Poolgrößen $\varrho \geq 1$ und Gruppengrößen $L \geq 2$ behandelt werden. Obige Forderungen (i) bis (iv) werden jedoch nur im Fall $\varrho = k \cdot L$ mit $k \geq L$ erfüllt. Andernfalls werden die Bedingungen typischerweise „umso schwerwiegender“ verletzt, je mehr Kopien von Lösungen in den Pool einzufügen sind.

Mit Hilfe des Gruppenbildungsgesetzes in Bemerkung 4.1 kann nun ein Verfahren formuliert werden, welches in $S \geq 1$ verschiedenen Stufen $t = 1, 2, \dots, S$ Mehrheitsläufe lokaler Suche einsetzt und in jeder Stufe t dieselbe Zahl ϱ von Läufen macht:

ALGORITHMUS 4.2

Seien $L \geq 2$ die Gruppengröße, $\varrho \geq 1$ die Poolgröße und $S \geq 1$ die maximale Stufe.

Schritt 1: Setze $t := 0$.

Mache ϱ zufällige Läufe lokaler Suche. Seien $x^0, x^1, \dots, x^{\varrho-1}$ die lokal maximalen Lösungen. Setze $\mathcal{P} := (x^0, x^1, \dots, x^{\varrho-1})$.

Schritt 2: Wähle zufällig eine Permutation $\pi \in S_\varrho$. Permutiere $\mathcal{P} = (x^0, x^1, \dots, x^{\varrho-1})$ zu $\mathcal{P} := (x^{\pi(0)}, x^{\pi(1)}, \dots, x^{\pi(\varrho-1)})$.

Schritt 3: Bilde aus der Menge $\{0, 1, \dots, \varrho-1\}$ „Gruppen“ $G_0, G_1, \dots, G_{\varrho-1}$ der Größe L gemäß Bemerkung 4.1.

Schritt 4: Für $i = 0, \dots, \varrho-1$ sei y^i die Lösung, die durch Mehrheitsbildung aus den durch G_i indizierten Lösungen hervorgeht. Mache ϱ Mehrheitsläufe lokaler Suche mit Startlösungen $y^0, y^1, \dots, y^{\varrho-1}$. Seien $z^0, z^1, \dots, z^{\varrho-1}$ die lokal maximalen Lösungen. Setze $\mathcal{P} := (z^0, z^1, \dots, z^{\varrho-1})$.

Schritt 5: Falls $t = S$, stopp. Gib eine insgesamt beste gefundene Lösung und ihren Zielfunktionswert aus.

Sonst setze $t := t + 1$ und gehe zu Schritt 2.

Der angegebene Algorithmus macht insgesamt $(S + 1) \cdot \varrho$ Läufe lokaler Suche. Man beachte, dass der aktuelle Pool in Schritt 2 permutiert wird, bevor die Lösungen für die Mehrheitsbildung in Gruppen eingeteilt werden. Praktische Versuche haben gezeigt, dass die Ausgaben auf diese Weise etwas besser sind als ohne Permutieren.

Algorithmus 4.2 lässt sich formal in die Klasse der *Genetischen Algorithmen* einordnen. Genetische Algorithmen wurden Mitte der Sechziger Jahre von HOLLAND entwickelt. Sie ahmen die natürliche Evolution nach und wenden genetische Operatoren wie *Rekombination*, *Selektion* und *Mutation* auf eine *Population* von *Individuen* an. Durch Veränderung

der Gene mittels dieser Operatoren soll die *Fitness* der Population gesteigert werden. Die wesentlichen Ideen zum Einsatz Genetischer Algorithmen als Optimierungs- und Suchtechnik in Ökonomie, Spieltheorie, Mustererkennung und maschinellem Lernen veröffentlichte HOLLAND 1975 [22]. Neben HOLLAND leistete GOLDBERG, ein Schüler HOLLANDs, wichtige Pionierarbeit auf dem Gebiet Genetischer Algorithmen. GOLDBERG gab 1989 [20] eine Liste praktischer Faustregeln für den Umgang mit Genetischen Algorithmen bei verschiedenen technischen Anwendungen an und verschaffte der noch jungen Klasse von Algorithmen eine breitere Popularität. Eine leicht verständliche und ausführliche Einführung in die Thematik findet man in [29].

Speziell in Algorithmus 4.2 sind die Individuen die Lösungen des Problems. Ihre Fitness wird durch den entsprechenden Zielfunktionswert ausgedrückt. Der Pool \mathcal{P} von Lösungen bildet die aktuelle Population von Individuen. Die Poolgröße ϱ heißt deshalb auch *Populationsgröße*. Zur Mutation wird lokale Suche eingesetzt. Die Einteilung der Individuen in Gruppen lässt sich als Elternwahl interpretieren. Die Gruppengröße L spielt die Rolle der Elternzahl. Der Elternwahl schließt sich die Zeugung von Nachkommen durch Rekombination der elterlichen Gene an. Als Rekombinationsoperator dient die Mehrheitsbildung. Eine Selektion findet statt, wenn aus den ϱ Individuen der Elterngeneration und ihren ϱ Nachkommen die neue Population gebildet wird. In dieser Arbeit wird die Elterngeneration vollständig durch die Nachkommen ersetzt.

4.2.1 Algorithmus 4.2 am Beispiel „Hanoi 5“

Im Folgenden wird der Frage nachgegangen, inwieweit sich f -Werte, Schrittzahlen und Fragenzahlen verändern, wenn statt des zweistufigen Mehrheitsbildungsverfahrens aus Abschnitt 4.1 der Genetische Algorithmus 4.2 verwendet wird. Dazu wird wieder das Benchmark „Hanoi 5“ herangezogen. Die entsprechenden Versuche wurden für die deterministisch bzw. die probabilistisch wiederholte zyklische schnelle lokale Suche sowie die wiederholte volle lokale Suche durchgeführt. Um die Ergebnisse mit Tabelle 4.1 vergleichen zu können, wurden die Parameter $L = 3$, $\varrho = 9$ und $S = 2$ beibehalten. Ferner wurden wiederum 2000 Läufe pro Variante lokaler Suche gemacht. Diese unterteilten sich in 74 unabhängige Aufrufe von Algorithmus 4.2 und zwei zusätzliche zufällige Läufe. Auf diese Weise ergeben sich insgesamt $74 \cdot ((S + 1) \cdot \varrho) + 2 = 2000$ Läufe. Bei allen drei Varianten wurden jeweils 2000 verschiedene lokal maximale Lösungen gefunden. Die Einzelergebnisse sind in Abbildung 4.7 sowie Tabelle 4.2 illustriert. Aus Platzgründen ist nur die Häufigkeitsverteilung bei der ersten der drei Varianten lokaler Suche dargestellt. Die Bedeutung der verwendeten Bezeichnungen ist wie in den analogen Abbildungen und Tabellen von Abschnitt 4.1. Im Vergleich mit den dortigen Daten stellt sich heraus, dass der Übergang vom Zweistufenverfahren zum Genetischen Algorithmus mit zwei Generationswechseln keine Einbußen bei den f -Werten oder der Effizienz des Verfahrens bewirkt.

Man kann nun die Populationsgröße ϱ , die Elternzahl L oder die höchste Generation S variieren. Allein durch Intuition geleitet, wird man alle drei Parameter möglichst groß wählen wollen: Eine hohe Generationenzahl wird allein schon durch die obigen Beobachtungen für den Fall $S = 2$ motiviert. Eine große Zahl ϱ von Individuen der Population

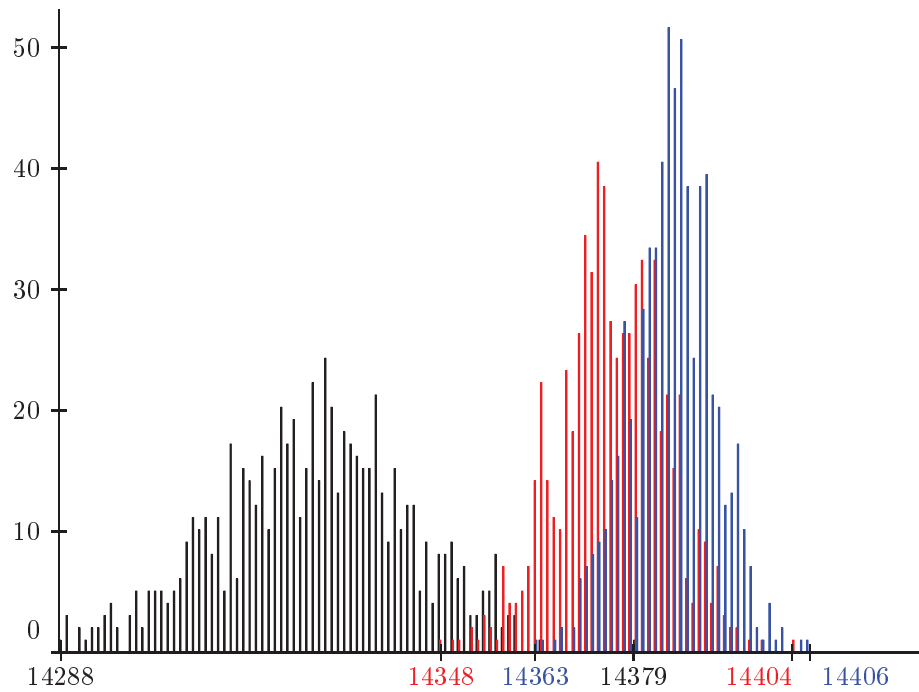


Abbildung 4.7: Häufigkeitsverteilung der gefundenen lokalen Maxima beim Genetischen Algorithmus mit $S = 2$, $\varrho = 9$, $L = 3$ und deterministisch wiederholter zyklischer schneller lokaler Suche für „Hanoi 5“

	\max_{zf}	\max_{Mh1}	\max_{Mh2}	\bar{f}_{zf}	\bar{f}_{Mh1}	\bar{f}_{Mh2}
dZS	14379	14404	14406	14328.0	14375.3	14384.6
pZS	14348	14392	14402	14301.9	14366.5	14382.1
V	14355	14403	14411	14305.7	14378.0	14393.5

	\bar{S}_{zf}	\bar{S}_{Mh1}	\bar{S}_{Mh2}	\bar{F}_{zf}	\bar{F}_{Mh1}	\bar{F}_{Mh2}
dZS	513.2	76.4	45.4	12898.3	6428.6	5416.6
pZS	505.1	84.6	56.6	10497.0	5803.4	4991.7
V	419.7	75.7	47.2	812320.8	148136.1	93007.0

Tabelle 4.2: Ergebnisse des Genetischen Algorithmus mit $S = 3$, $\varrho = 9$, $L = 3$ und verschiedenen Varianten lokaler Suche bei „Hanoi 5“

sorgt für eine große Vielfalt, d.h. eine hohe *Diversität*, des genetischen Materials. Eine hohe Diversität wiederum beugt verfrühter Stagnation des Verfahrens durch „Inzuchteffekte“ vor. Eine große Elternzahl L erscheint aufgrund der Beobachtungen in Abschnitt 2.2.1 intuitiv gerechtfertigt: Die „Wachstumsgeschwindigkeit“ der Erfolgskurven von Max-3-SAT gegen 1 nimmt mit wachsender Gruppengröße zu, die Misserfolgstäler werden flacher. In Abschnitt 4.2.2 wird die Frage der Parameterwahl ausführlich untersucht. Es wird sich zeigen, dass große Elternzahlen trotz dieser Argumente nicht ratsam sind.

In einem weiteren Versuch zu „Hanoi 5“ sei $S = 9$, $\varrho = 20$ und $L = 3$. Algorithmus 4.2 wird zehnmal unabhängig gestartet. Daraus ergeben sich 2000 Läufe lokaler Suche. Zum Einsatz kommt die deterministisch wiederholte zyklische schnelle lokale Suche ohne Sidesteps.

Die Ergebnisse sind Tabelle 4.3 zu entnehmen. Mit t wird die Generation der aktuellen Population bezeichnet. Für $t = 0, 1, \dots, 9$ ist \max_t der größte f -Wert der 200 lokal maximalen Lösungen in Generation t und \bar{f}_t ihr Durchschnittswert. \bar{S}_t bzw. \bar{F}_t kennzeichnet die im Mittel benötigte Schritt- bzw. Fragenzahl der Läufe lokaler Suche, die zu den Lösungen in Generation t führten.

t	\max_t	\bar{f}_t	\bar{S}_t	\bar{F}_t
0	14364	14326.4	514.2	12604.9
1	14397	14373.3	73.8	6247.9
2	14400	14384.3	48.8	5487.1
3	14399	14386.1	30.0	5005.2
4	14402	14386.1	16.5	4632.7
5	14398	14385.9	9.5	4128.8
6	14400	14385.8	5.3	3731.3
7	14403	14385.6	2.7	3253.0
8	14403	14385.5	1.6	2871.3
9	14402	14385.7	0.6	2332.0

Tabelle 4.3: *Ergebnisse des Genetischen Algorithmus mit $S = 9$, $\varrho = 20$, $L = 3$ und deterministisch wiederholter zyklischer schneller lokaler Suche bei „Hanoi 5“*

Man erkennt, dass die Schritt- und Fragenzahlen von Generation zu Generation abnehmen. $\bar{S}_9 < 1$ zeigt, dass einige Startlösungen für die Läufe zur letzten Generation lokal maximal waren. Tatsächlich waren 230 der insgesamt 1800 Startlösungen der Mehrheitsläufe lokal maximal. 39 Vektoren davon stimmten mit einer der zur Mehrheitsbildung herangezogenen Lösungen überein. Insgesamt wurden in den 2000 Läufen 1934 verschiedene lokal maximale Lösungen gefunden. In den letzten Generationen ist also die Diversität der entsprechenden Population schon deutlich zurückgegangen. Insbesondere würde ein Hochsetzen der Schranke S kaum noch sinnvoll sein. Ein Blick auf die Werte \bar{f}_t zeigt, dass der Algorithmus schon nach vier Generationswechseln hinsichtlich der durchschnittlichen f -Werte keine Fortschritte mehr machte. Trotzdem traten erst in den Generationen 7 und 8 die insgesamt besten f -Werte von 14403 auf. Die beiden zugehörigen lokal maximalen Lösungen waren verschieden. Aus Sicht der Optimierung handelte es sich um „glückliche Ausreißer“.

Es sollen nun zusätzlich Sidesteps erlaubt sein. Wie in den Abschnitten 2.1.1 und 4.1 wird die Schrankenmethode verwendet: Zwischen je zwei echten Verbesserungen während eines Laufs lokaler Suche sind höchstens 1931 (= Zahl der Variablen) Sidesteps zulässig. Alle weiteren Größen und Parameter seien wie oben. Die neuen Ergebnisse sind in Tabelle 4.4 dargestellt. Dass die erzielten f -Werte deutlich besser sind als ohne Erlaubnis von Sidesteps, ist nicht weiter verwunderlich. Man beachte, dass das globale Maximum bei „Hanoi 5“ die Zahl der Klauseln ist, also 14468. Der beste gefundene f -Wert 14463 geht also relativ knapp am Optimum vorbei. 23 der 1800 Startlösungen der Mehrheitsläufe waren lokal maximal. All diese Mehrheitsvektoren stimmten mit einer der zur Mehrheitsbildung herangezogenen Lösungen überein. Insgesamt wurden in den 2000 Läufen 1977 verschiedene lokal maximale Lösungen gefunden. Die durchschnittlichen Zielfunk-

t	\max_t	\bar{f}_t	\bar{S}_t	\bar{F}_t
0	14450	14440.9	4361.5	291626.1
1	14457	14448.2	3396.7	297432.4
2	14463	14454.0	3393.7	332639.3
3	14462	14456.8	3352.4	336711.5
4	14463	14457.2	3261.8	322893.4
5	14462	14457.2	3236.2	317386.2
6	14462	14457.1	3167.6	314948.9
7	14462	14456.9	3122.7	309460.9
8	14461	14456.8	3062.6	301954.9
9	14461	14456.8	2950.5	293917.4

Tabelle 4.4: *Ergebnisse des Genetischen Algorithmus mit $S = 9$, $\varrho = 20$, $L = 3$ und deterministisch wiederholter zyklischer schneller lokaler Suche mit Erlaubnis von Sidesteps bei „Hanoi 5“*

tionswerte steigerten sich bis Generation 4. Trotzdem lässt sich selbst in Generation 9 noch keine Stagnation der Schritt- und Fragenzahlen erkennen. Man beachte dazu, dass in die Schrittzahlen auch die Sidesteps mit eingehen. Ab einem gewissen Zielfunktionsniveau bewegt sich lokale Suche fast nur noch auf Plateaus gleicher f -Werte. Doch dieser Effekt verstärkt wenigstens die Mutation und erhöht die Diversität der Population.

Es gibt auch verschiedene Möglichkeiten, die Diversität der Population künstlich zu erhöhen. Beispielsweise kann man einzelne Individuen durch zufällig bzw. willkürlich gewählte Lösungen ersetzen. Naheliegender ist auch, gute Lösungen aus anderen Algorithmen mit entsprechend „anders strukturierten Genen“ zu übernehmen. Eine weitere Möglichkeit besteht darin, zwischendurch Mutationen der Individuen in Form von zufälligem Flippen einzelner Komponenten mit einer bestimmten Wahrscheinlichkeit zu erlauben. Für die Zwecke der vorliegenden Arbeit wird auf derartige Techniken verzichtet. Hier geht es mehr darum, den Einfluss der Mehrheitsbildung zu untersuchen.

In der Literatur wird die gezielte Mutation durch lokale Suche oder andere Heuristiken meist von der zufälligen Mutation getrennt. Man spricht dann bei zusätzlichem Einsatz einer Heuristik von *Hybriden Genetischen Algorithmen*. Im Fall lokaler Suche werden Hybride Genetische Algorithmen auch als Verfahren *Genetischer (lokaler) Suche* bezeichnet.

Algorithmus 4.2 wurde auf etliche Instanzen zu Max-SAT, dem Shiftregister-Problem und dem Ising-Spینگlas-Problem auf dem Torus angewendet. Das Verfahren zeigte jeweils ein analoges Verhalten, wie hier beschrieben. An dieser Stelle wird daher auf eine Darstellung weiterer Daten verzichtet.

4.2.2 Populationsgröße, Generationenzahl und Elternzahl

Im Folgenden werden Untersuchungen zur Wahl von Populationsgröße ϱ und Elternzahl L angestellt. Populationsgröße ϱ und optimale Höchstzahl S an Generationen sind insofern

gekoppelt, als große ϱ -Werte für eine höhere Diversität der Population sorgen. Das Verfahren stagniert dann in einer etwas späteren Generation.¹ In den folgenden Versuchen wurde S stets hinreichend groß gewählt, so dass der Algorithmus nicht verfrüht, d.h. vor seiner Stagnation, abgebrochen wurde. Dazu wurde 100 Generationen lang gerechnet.

Für erste Versuche wurden zufällig erzeugte Instanzen des Shiftregister-Problems herangezogen. Die zufällige Wahl einer Instanz erfolgte genau so, wie sie zu Beginn von Abschnitt 2.2.3 beschrieben worden war. Als Dimension des Lösungsraums wurde $n = 1000$ festgehalten. Ferner wurde die Zahl L der Eltern fest aus $\{2, 3, \dots, 9\}$ gewählt. Für jede Poolgröße $\varrho \in \{1, 2, \dots, 100\}$ wurden 1000 Instanzen zufällig und unabhängig erzeugt. Nach Wahl einer Instanz wurde Algorithmus 4.2 aufgerufen. Dieser berechnete 100 Generationen (d.h. $S = 99$) zu je ϱ Individuen. Dabei wurde die probabilistisch wiederholte zyklische schnelle lokale Suche eingesetzt. Da es sich beim Shiftregister-Problem um ein Minimierungsproblem handelt, wurde in jedem Lauf eine lokal *minimale* Lösung ermittelt. Das Verfahren merkte sich den besten Defekt, d.h. den kleinsten f -Wert, aus den insgesamt $100 \cdot \varrho$ Läufen. Am Ende wurde über die besten Defekte der 1000 Instanzen gemittelt. In Abbildung 4.8 sind die durchschnittlichen besten Defekte in Abhängigkeit von ϱ für jedes feste $L \in \{2, 3, \dots, 9\}$ skizziert und mit Linien verbunden.

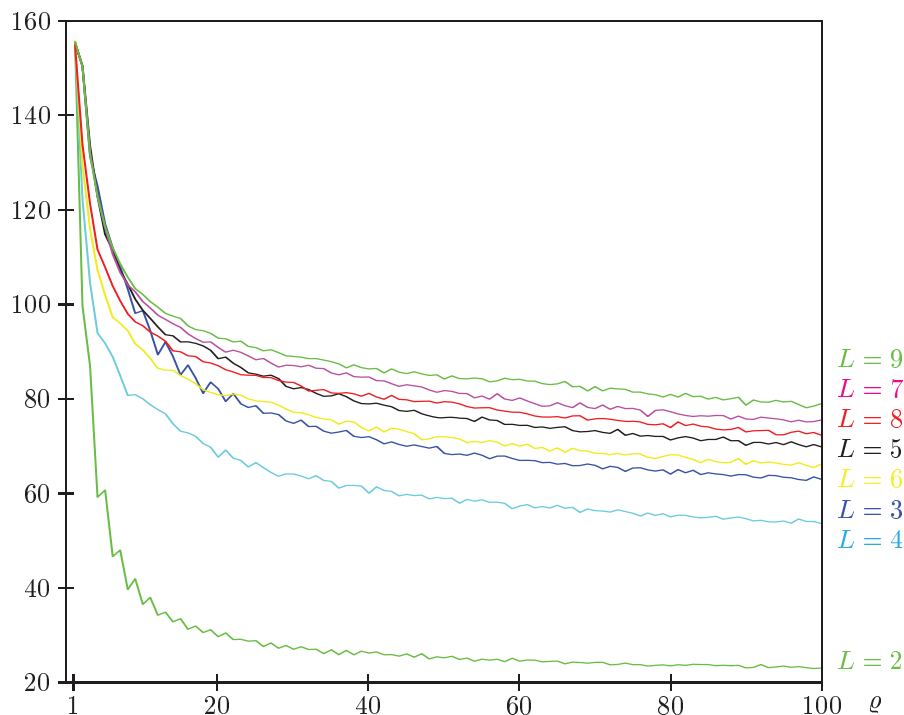


Abbildung 4.8: **(Shiftregister-Problem)** Durchschnittlich erzielte kleinste Defekte in Abhängigkeit von Populationsgröße ϱ und Elternzahl L unter $n = 1000$, $S = 99$ und probabilistisch wiederholter zyklischer schneller lokaler Suche bei Mehrheitsbildung

Wie erwartet, werden die besten gefundenen f -Werte im Durchschnitt mit wachsender Populationsgröße ϱ kleiner. Dies gilt für jede fest gewählte Elternzahl L . Überraschend ist jedoch, dass bei festgehaltener Populationsgröße ϱ die mittleren besten Defekte nicht

¹Beispielsweise kann das Unterschreiten der durchschnittlichen Schrittzahl von lokaler Suche ohne Sidesteps zur aktuellen Population unter den Wert 1.0 als Stagnation bezeichnet werden.

mit wachsender Elternzahl L kleiner werden. Vielmehr sprechen die Daten für eine Verwendung kleiner, gerader Elternzahlen. Mit großem Vorsprung gegenüber allen anderen Gruppengrößen liefert Elternzahl 2 die besten Defekte. Zwei Elternteile zur Erzeugung der Nachkommen heranzuziehen (*bisexuelle Rekombination*), erweist sich also nach dem Vorbild der natürlichen Evolution als die optimale Wahl. Die Optimalität von $L = 2$ lässt sich folgendermaßen erklären: Bei geraden Elternzahlen ist der Nachkomme, d.h. der Mehrheitsvektor, nicht unbedingt eindeutig. Eine Mehrdeutigkeitsstelle $i \in \{1, 2, \dots, n\}$ tritt auf, wenn in den i -ten Komponenten der elterlichen Individuen $\frac{L}{2}$ Einsen und $\frac{L}{2}$ Nullen vorkommen. Dieses Ereignis ist bei $L = 2$ am wahrscheinlichsten. Liegt eine Mehrdeutigkeitsstelle i vor, wird bei der Mehrheitsbildung eine der Alternativen 0 oder 1 für die i -te Komponente des Mehrheitsvektors zufällig gewählt. Dieses „Auswürfeln“ des Eintrags trägt wesentlich zur Diversitätserhöhung der Population bei.

Analoge Versuche wurden auch für das Dynamische Optimierungsproblem mit Problemparameter $n = 10\,000$ durchgeführt. Eine zufällig gewählte Instanz besteht aus Matrizen $C(t) = (c_{ij}(t))_{i,j=0,1}$ für $t = 1, \dots, n - 1$ mit zufällig gewählten Komponenten in $[0, 1[$. Als Poolgrößen wurden $\varrho = 1, 2, \dots, 20$ herangezogen. Alle anderen Größen waren wie oben. Man beachte allerdings, dass das Dynamische Optimierungsproblem ein Maximierungsproblem darstellt. Die Ergebnisse sind in Abbildung 4.9 illustriert. Die „Kurven“ sprechen auch bei diesem Problem für kleine gerade Elternzahlen. Wieder ist Elternzahl 2 optimal.

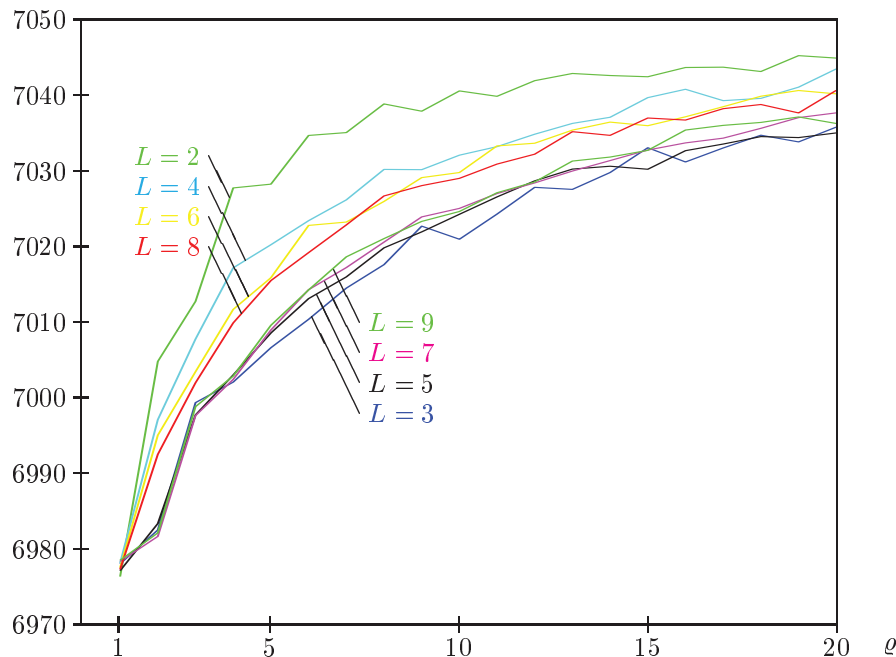


Abbildung 4.9: **(Dynamisches Optimierungsproblem)** Durchschnittlich erzielte größte lokale Maxima in Abhängigkeit von Populationsgröße ϱ und Elternzahl L unter $n = 10\,000$, $S = 99$ und probabilistisch wiederholter zyklischer schneller lokaler Suche bei Mehrheitsbildung

In Anlehnung an die natürliche Evolution sind in der Literatur Rekombinationen mit mehr als zwei Elternteilen (*multisexuelle Rekombination*) bis heute die Ausnahme geblieben. In einigen wenigen Arbeiten sind jedoch Experimente vorgestellt worden, in denen sich multisexuelle Rekombinationsoperatoren als vorteilhaft erwiesen hatten. Beispiele hierfür

sind *Gene Scanning* oder *Diagonal Crossover* [14]. Gene Scanning ist eine multisexuelle Verallgemeinerung des bisexuellen *Uniform Crossover*, während Diagonal Crossover den *N Point Crossover* auf mehr als zwei Eltern überträgt.² Versuche haben gezeigt, dass je nach Optimierungsproblem und Art der Rekombination Elternzahlen von 2 bis hin zu 10 oder mehr zu den besten Ergebnissen führen können [14].

4.2.3 Rekombination mit Proporz- und Einstimmigkeitsbildung

Analog zur Mehrheitsregel wurden auch die Proporzregel sowie die Einstimmigkeitsregel aus Abschnitt 3.1 zur Rekombination im Genetischen Algorithmus 4.2 eingesetzt. Alle weiteren verwendeten Größen und Rahmenbedingungen waren wie im vorangegangenen Abschnitt. Es war also nur die Bildungsregel in Schritt 4 des Algorithmus abzuändern. Man beachte, dass bei Elternzahl 2 alle drei Bildungsregeln identisch sind. In den folgenden Darstellungen zur Proporz- bzw. Einstimmigkeitsregel mit Elternzahl 2 wurden daher die Daten aus den Versuchen mit der Mehrheitsregel übernommen.

Betrachte zunächst die Ergebnisse der Proporzbildung für das Shiftregister-Problem (Abbildung 4.10). Hier sind die durchschnittlich erreichten besten Defekte in Abhängigkeit der

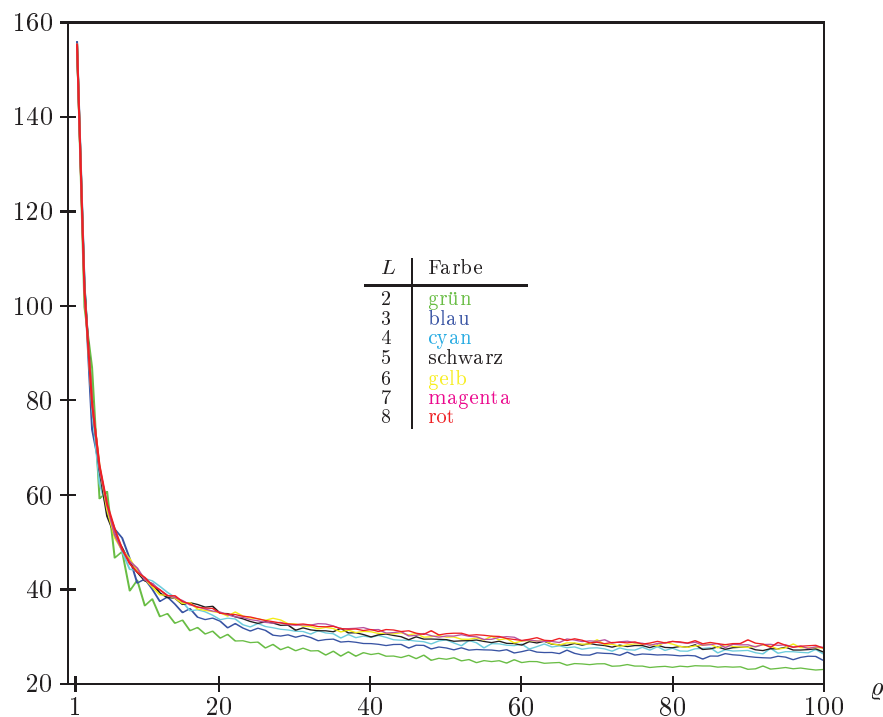


Abbildung 4.10: **(Shiftregister-Problem)** Durchschnittlich erzielte kleinste Defekte in Abhängigkeit von Populationsgröße q und Elternzahl L unter $n = 1000$, $S = 99$ und probabilistisch wiederholter zyklischer schneller lokaler Suche bei Proporzbildung

Populationsgröße q skizziert. Als Elternzahlen wurden $L = 2, 3, \dots, 8$ untersucht. Der Graphik ist zu entnehmen, dass wie schon bei der Mehrheitsbildung auch bei Proporzbildung

²zu den Begriffen siehe etwa [29]

Elternzahl 2 optimal ist. Ferner scheinen die durchschnittlich erreichten besten Defekte bei fester Populationsgröße ϱ mit wachsender Elternzahl L schlechter zu werden. Ein Vergleich mit den Daten der Mehrheitsbildung zeigt, dass bei jedem hinreichend großen ϱ die Werte der Proporzbildung für jedes $L \geq 3$ besser sind als die Werte der Mehrheitsbildung.

Abbildung 4.11 zeigt die analogen Ergebnisse für das Dynamische Optimierungsproblem. Im Unterschied zum Shiftregister-Problem kann hier keine Überlegenheit einer be-

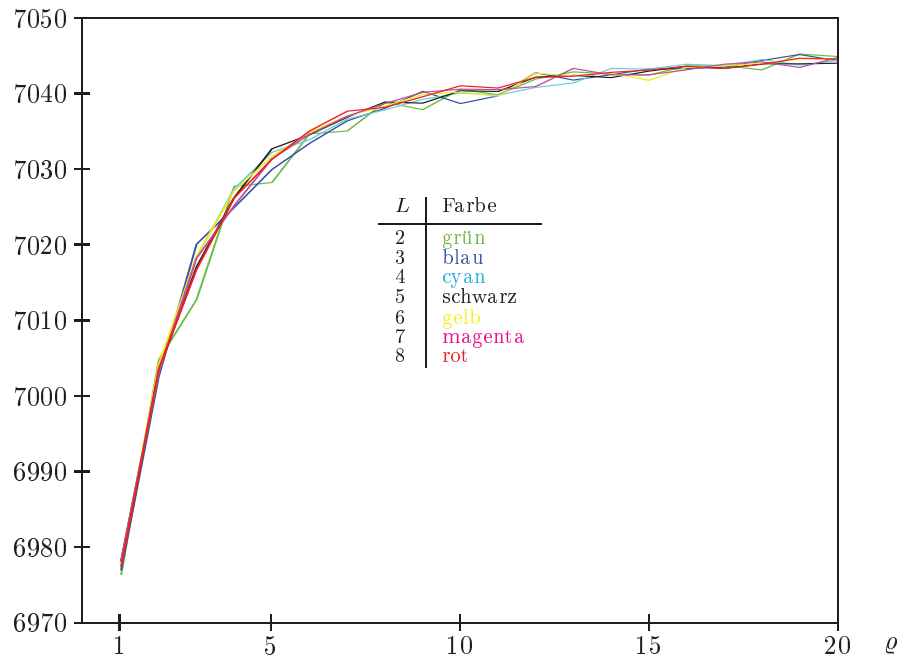


Abbildung 4.11: **(Dynamisches Optimierungsproblem)** Durchschnittlich erzielte größte lokale Maxima in Abhängigkeit von Populationsgröße ϱ und Elternzahl L unter $n = 10\,000$, $S = 99$ und probabilistisch wiederholter zyklischer schneller lokaler Suche bei Proporzbildung

stimmten Elternzahl festgestellt werden. Aber zumindest stellt sich im Einklang mit dem Shiftregister-Problem heraus, dass die durchschnittlich erzielten f -Werte für jedes hinreichend große ϱ und jedes $L \geq 3$ besser sind als bei Verwendung der Mehrheitsregel.

In Abbildung 4.12 sind die Ergebnisse der Einstimmigkeitsbildung für das Shiftregister-Problem dargestellt. Ist $\varrho \in \{1, 2, \dots, 6\}$ fest, so werden die gefundenen kleinsten Defekte mit wachsender Elternzahl besser. Insbesondere liefert entgegen den Beobachtungen bei der Mehrheitsbildung und der Proporzbildung Elternzahl 2 bei der Einstimmigkeitsbildung die schlechtesten Werte. Die Begründung erfolgt mit einem Diversitätsargument: Je größer L , desto geringer ist die Wahrscheinlichkeit, dass die i -ten Komponenten $x_i^1, x_i^2, \dots, x_i^L$ von L Lösungen $x^1, x^2, \dots, x^L \in \{0, 1\}^n$ übereinstimmen. Mit L nimmt also die Wahrscheinlichkeit zu, dass eine Komponente des Einstimmigkeitsvektors per Zufallsentscheid auf 0 bzw. 1 gesetzt wird. Zufallsentscheide tragen wesentlich zur Diversitätserhöhung bei.

Ab $\varrho = 7$ tritt folgendes Phänomen auf: Die „Kurven“ der durchschnittlichen besten Defekte „brechen“ — beginnend mit den großen Elternzahlen L — nacheinander „weg“. Diese

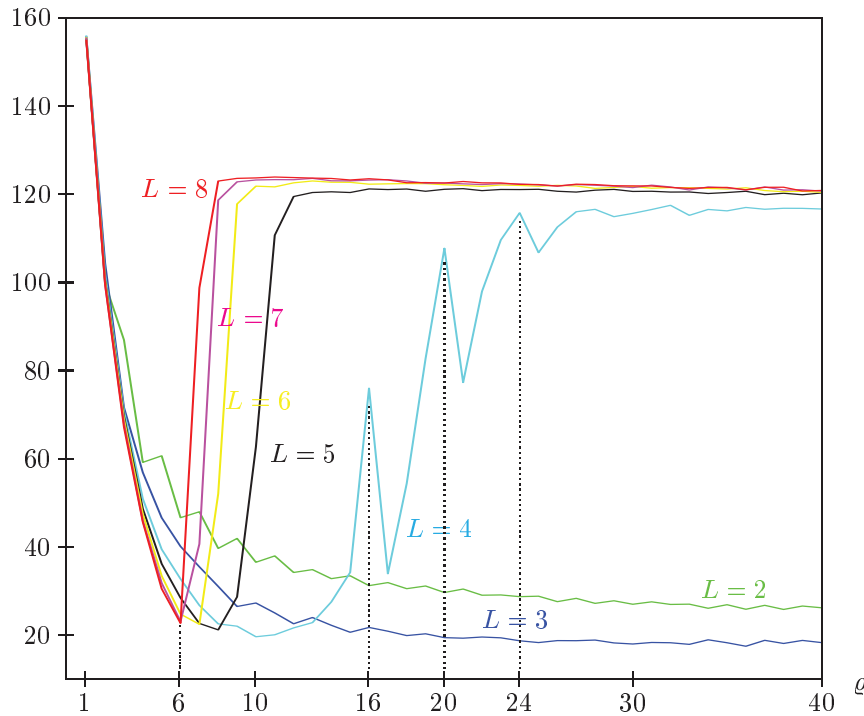


Abbildung 4.12: **(Shiftregister-Problem)** Durchschnittlich erzielte kleinste Defekte in Abhängigkeit von Populationsgröße ρ und Elternzahl L unter $n = 1000$, $S = 99$ und probabilistisch wiederholter zyklischer schneller lokaler Suche bei Einstimmigkeitsbildung

Beobachtung lässt sich für jedes $L \geq 4$ machen. Für hinreichend große Populationsgrößen ρ pegeln sich die „Kurven“ dann bei einem bestimmten f -Wert-Niveau in der Nähe von 120 ein. Eine mögliche Erklärung hierfür ist die folgende: Mit der Populationsgröße ρ nimmt die Diversität der Population einer bestimmten Generationenzahl zu. Die Wahrscheinlichkeit, dass die i -ten Komponenten $x_i^1, x_i^2, \dots, x_i^L$ von L Lösungen $x^1, x^2, \dots, x^L \in \{0, 1\}^n$ übereinstimmen, sinkt bis hin zu einer gewissen (kleinen) Restwahrscheinlichkeit $p_i(\rho)$. Mit Wahrscheinlichkeit $q_i(\rho) = 1 - p_i(\rho)$ wird die i -te Komponente des Einstimmigkeitsvektors zufällig bestimmt. Ist $q_i(\rho)$ nahe bei 1, stimmt der beste gefundene f -Wert aus den $100 \cdot \rho$ Läufen praktisch mit dem besten f -Wert aus $100 \cdot \rho$ zufälligen Läufen überein. Ein entsprechender Versuch für $\rho = 20$ bestätigte, dass der beste f -Wert aus $100 \cdot 20$ zufälligen Läufen im Durchschnitt, d.h. gemittelt über 1000 zufällig und unabhängig gewählte Instanzen, knapp über 120 liegt.

Interessant ist ferner das „Zickzackverhalten“ der „Kurve“ für $L = 4$. An den Stellen $\rho \in \{16, 20, 24\}$ sind die Werte lokal maximal, bei $\rho \in \{17, 21, 25\}$ lokal minimal. Dieser Effekt ist auf die Gruppenbildung in Bemerkung 4.1 zurückzuführen. Für obige lokale Maximalstellen ρ gilt nämlich $\rho = k \cdot L$ mit $k \geq L$. Aufgrund dieses Zusammenhangs zwischen ρ und L sind zur Einstimmigkeitsbildung keine Kopien von Individuen nötig. Die Wahrscheinlichkeit $q_i(\rho)$ ist dann bzgl. ρ lokal maximal.

Als weiterer Grund für das „Wegbrechen“ der „Kurven“ für $L \geq 4$ ist eine zu geringe Generationenzahl denkbar. Neben den Zufallsentscheiden bei Einsatz der Einstimmigkeitsbildung bewirken nämlich gerade die hohen Populationsgrößen eine relativ späte Stagnation

des Algorithmus. Analog zu $S = 99$ wurden die Versuche deshalb für $S = 499$ durchgeführt. Für jede der 1000 Instanzen pro ϱ und L wurde also 500 Generationen lang gerechnet. Hier ergaben sich jedoch praktisch die gleichen Ergebnisse. Auf eine Darstellung der Daten wird daher verzichtet.

Beim Shiftregister-Problem wurden einige weitere Versuche mit Mehrheitsbildung, Proportionalbildung und Einstimmigkeitsbildung gemacht, und zwar mit Flippen und Drehen lokal minimaler Lösungen (vgl. Abschnitt 2.2.3). Die Unterschiede zu den Anwendungen der entsprechenden Regeln ohne Flippen und Drehen waren jedoch gering. Sie sind im Hinblick auf die Unterschiede, die sich durch Variation der Bildungsregel ergeben, nicht nennenswert.

Betrachte schließlich Abbildung 4.13. Sie illustriert die Ergebnisse des Genetischen Algorithmus 4.2 mit der Einstimmigkeitsregel, angewendet auf das Dynamische Optimierungsproblem. Es lässt sich erkennen, dass alle „Kurven“ zu $L \geq 4$ im Wesentlichen einen

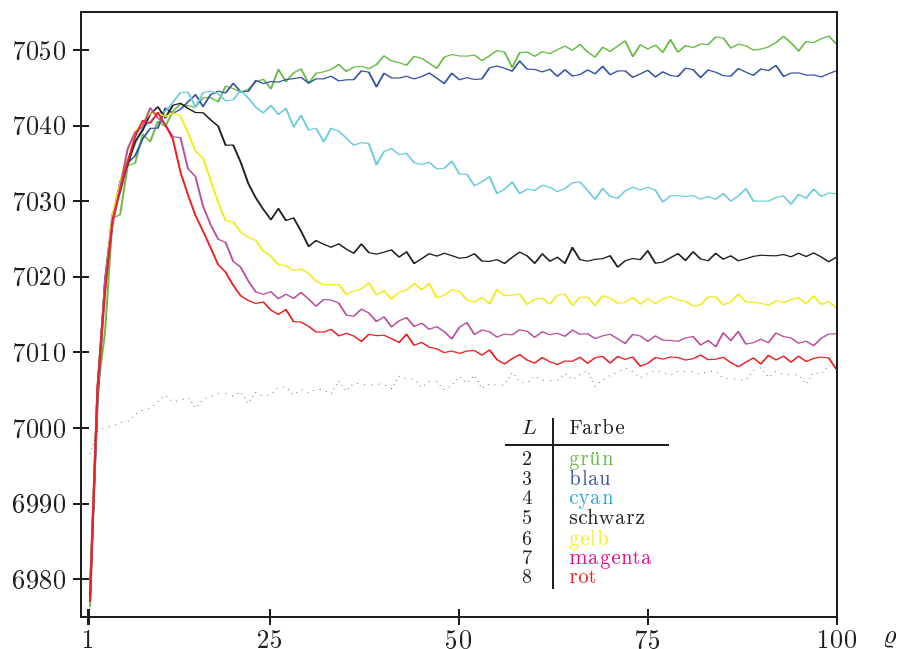


Abbildung 4.13: **(Dynamisches Optimierungsproblem)** Durchschnittlich erzielte größte lokale Maxima in Abhängigkeit von Populationsgröße ϱ und Elternzahl L unter $n = 10\,000$, $S = 99$ und probabilistisch wiederholter zyklischer schneller lokaler Suche bei Einstimmigkeitsbildung

unimodalen Verlauf zeigen. Ab einer bestimmten Populationsgröße steigen nur noch die zu den kleinen Elternzahlen gehörenden „Kurven“ weiter an. Bei kleinen Populationsgrößen führt Elternzahl 2 dagegen zu vergleichsweise schlechten f -Werten. Die Beobachtungen beim Dynamischen Optimierungsproblem entsprechen also im Wesentlichen den Erkenntnissen beim Shiftregister-Problem. Die gestrichelte „Kurve“ zeigt für jede Populationsgröße ϱ die größten erzielten lokalen Maxima aus $100 \cdot \varrho$ zufälligen Läufen, gemittelt über 10 000 Instanzen. Sie belegt, dass sich Einstimmigkeitsvektoren bei großen Elternzahlen und Populationsgrößen kaum noch von zufälligen Vektoren unterscheiden.

Anhang

Mittelwertbildung

bei einem Scheduling-Benchmark und Varianten

Am 6. September 1997 wurde in der holländischen Zeitschrift *De Telegraaf* zum zweiten Mal nach 1996 zur Teilnahme an einem Mathematik-Wettbewerb unter dem Titel „Whizzkids Mathematics Contest“ aufgerufen. J. K. LENSTRA, E. AARTS und ihre Mitarbeiter-Teams hatten ein Benchmark-Problem entworfen, bei dem es darum ging, einen Elternsprechabend an einer Schule mit 15 Lehrern und 20 Elternpaaren zu organisieren. Parallel zum Aufruf in *De Telegraaf* wurden die Ausgangsdaten im Internet¹ veröffentlicht. Im Prinzip war weltweit jeder teilnahmeberechtigt. Die Ergebnisse von „Professionals“, d.h. in mathematischer Lehre und Forschung tätigen Bewerbern, wurden jedoch getrennt ausgewertet. Vorrangiges Ziel war, bei Schülern Spaß und Interesse an Mathematik zu wecken und pfiffige Ideen zu sammeln. Abbildung A.1 zeigt die Original-Datentabelle der Internet-Seite.

De opgave Print deze pagina zó uit dat hij op een liggend A-4 past *WHIZZKIDS '97 prijsvraag*

Ouders	Aanvraag								
1	(E16)	(B20)	(F12)	(G39, M13, N10)	(P13)				
2	(A22)	(G27)	(C20)	(D25, E32, F8)	(B14, H19, J9)	(K29, M29)	(N26, P5)		
3	(B9)	(D9, E15)	(G29, H52)	(M24, J29)	(N13, P19)	(R27)			
4	(K30, B52)	(C27)	(D19, E32)	(F16, G10, H8)	(J18, A12)	(L23)	(M24, N26)	(P41)	
5	(A16)	(B18)	(C17, M33, E18)	(H56, L13)	(N22, P7)	(R13)			
6	(B12, D7)	(C27, F21)	(G12, H6)	(K73)	(L30)	(M22)	(N10, R32)		
7	(B6)	(C32)	(J18)	(H55, L9)	(N24)	(P48)	(R29)		
8	(A18)	(C27, D60)	(F21)	(J31)	(G22, K26)	(M13)	(P16, R26)		
9	(A34)	(B18)	(C9, K29)	(M20)	(P12)				
10	(A30)	(C15, D11)	(E15)	(G35, J36)	(L23)	(M17)	(P18, R10)		
11	(A30)	(D17)	(E8, F12)	(G12, H15, J34)	(L11, N13)	(R26)			
12	(A38, B14)	(G31)	(D11)	(F13, C19, J11)	(K12)	(L30, N12)	(R30)		
13	(A30, D55, E31)	(F18)	(K17, J25)	(H15)	(L33)	(P13, R8)			
14	(C15, B12, E26)	(F62, H14, J25)	(K29, L16)	(M13, N9)	(P15)	(R30)			
15	(A28, B45)	(C22, F10, D23)	(H16, J18)	(K12)	(L31, N24)				
16	(B23)	(C15)	(D21)	(F14, H21, K15)	(L16)	(N13, R23)			
17	(C10)	(F9)	(G37)	(H19, E10, P18)	(R10)				
18	(B16, L35)	(D14, E74, F21)	(G27)	(J13)	(K14)	(N17, P8)			
19	(B23, C23)	(D25)	(A36)	(G14, K11)	(M89, N71)	(P15, R15)			
20	(B17, C20)	(E18)	(J27, F59)	(L28)	(N8)	(P42, R18)			

Abbildung A.1: Datentabelle zum Scheduling-Benchmark „Whizzkids '97“

Jede Zeile der Tabelle wird in der ersten Spalte von der Nummer i des i -ten Elternpaares, $i = 1, \dots, 20$, angeführt. Die weiteren Spalten der i -ten Zeile enthalten die Lehrer, die Elternpaar i sprechen möchte, sowie die zugehörigen Gesprächsdauern. Die 15 Lehrer sind dabei mit den Großbuchstaben A, B, C, D, E, F, G, H, J, K, L, M, N, P, R symbolisiert. Der Eintrag (E16) in Zeile 1 und Spalte 2 bedeutet beispielsweise, dass das erste Elternpaar

¹erhältlich unter <http://www.win.tue.nl/whizzkids/1997/index.html>

Lehrer E genau 16 Minuten lang sprechen möchte. In jeder Zeile müssen die Sprechtermine aus Spalte s vor den Terminen aus Spalte s' stattfinden, falls $2 \leq s < s' \leq 9$. In welcher Reihenfolge die Termine *innerhalb* einer Spalte s abgehandelt werden, spielt keine Rolle. Elternpaar 12 etwa darf die Lehrer F, C und J in beliebiger Reihenfolge sprechen. Alle drei Termine müssen aber *nach* Lehrer D und *vor* Lehrer K stattfinden. Ferner dürfen sich Sprechtermine eines Lehrers mit verschiedenen Elternpaaren nicht überschneiden. Sprechtermine dürfen auch nicht unterbrochen werden. Übergangszeiten zwischen Terminen sind zu vernachlässigen, d.h. zwei Termine eines Elternpaares oder eines Lehrers dürfen sich nahtlos aneinanderreihen. Der Sprechabend beginnt pünktlich um 16:00 Uhr.

Es sollte nun ein Terminplan aufgestellt werden, der die Restriktionen sämtlich erfüllt. Unter den Bezeichnungen a_i für den Beginn des ersten Termins von Elternpaar $i \in \{1, \dots, 20\}$ sowie e_i für das Ende seines letzten Termins sollte der Plan zudem folgende zwei Werte minimieren:

$$(1) \quad \max \{e_i \mid i = 1, \dots, n\}$$

$$(2) \quad \sum_{i=1}^n (e_i - a_i)$$

Der Sprechabend sollte also möglichst bald beendet werden können, und die Elternpaare sollten sich durchschnittlich so kurz wie möglich in der Schule aufhalten müssen. Beide Kriterien waren gleichzeitig zu beachten, wobei (1) Priorität hatte. Bei gleichen Werten für (1) entschied Kriterium (2). Die Firma CMG aus Eindhoven sponserte den Wettbewerb mit diversen Siegprämien. Lösungsvorschläge waren bis 21. Oktober '97 einzureichen. Insgesamt gingen 343 Teilnahmen ein.² Die beste Lösung wurde von P. HOOGENDIJK, einem „Professional“ aus Eindhoven, gefunden. Sein Terminplan hatte eine Gesamtdauer von 469 Minuten und eine Gesamtverweildauer der Elternpaare von 5241 Minuten. Später wiesen die Teams von LENSTRA und AARTS mit Branch-and-Bound-Methoden nach, dass 469 eine untere Schranke für Kriterium (1) darstellt.

Formal handelt es sich bei „Whizzkids '97“ unter Optimierungskriterium (1) um eine Instanz eines verallgemeinerten *Job-Shop-Scheduling-Problems* (vgl. [15]). GAREY, JOHNSON und SETHI [16] haben 1976 die NP-Vollständigkeit des zu Job-Shop Scheduling gehörenden Entscheidungsproblems nachgewiesen. „Whizzkids '97“ ist folglich Instanz eines mindestens so schwierigen Problems.

Im weiteren Verlauf dieser Arbeit werden das Whizzkids-Problem sowie bestimmte Variationen desselben Typs behandelt. Eine Instanz besteht aus einer Datentabelle obiger Form mit $n \geq 1$ Eltern, $m \geq 1$ Lehrern und k Sprechwünschen. Jedes Elternpaar hat mindestens einen Sprechwunsch. Innerhalb der Sprechwünsche eines Elternpaares sind keine Wiederholungen von Lehrern enthalten. Die Lösungen sind alle zulässigen Schedules, d.h. die Terminpläne, die alle geforderten Restriktionen erfüllen. In Anlehnung an die verwendete Datenstruktur wird ein **Termin** definiert als Quintupel

$$(i, j, d, a, e) \in T := \{1, \dots, n\} \times \{1, \dots, m\} \times \mathbb{N}^+ \times \mathbb{N}_0 \times \mathbb{N}^+.$$

²Unter <http://www.win.tue.nl/whizzkids/1997/overview.txt> ist eine Hitliste der besten eingereichten Lösungen erhältlich.

DEFINITION A.1 (Zulässiger Schedule)

Ein **zulässiger Schedule** ist ein k -Tupel $S = (t_1, \dots, t_k) \in T^k$ mit $t_p = (i_p, j_p, d_p, a_p, e_p)$ für $p = 1, \dots, k$, das folgende Bedingungen erfüllt:

- (i) Jeder Termin t_p , $p = 1, \dots, k$, repräsentiert genau einen Sprechwunsch der Datentabelle. Dabei ist i_p das Elternpaar, j_p die betreffende Komponente der Enumeration (A, B, C, \dots, R) der Lehrer, d_p die Dauer, a_p der Anfangszeitpunkt und e_p der Endzeitpunkt des Termins t_p . Als Zeiten gelten dabei jeweils die Anzahl an Minuten, gerechnet ab 16:00 Uhr. i_p , j_p und d_p sind per Eingabe bekannt. a_p variiert je nach Schedule S . Für den Endzeitpunkt gilt $e_p = a_p + d_p$.
- (ii) Sind $p, q \in \{1, \dots, k\}$ mit $i_p = i_q$ und sind die Spaltennummern s_p bzw. s_q der Datentabelle, in denen t_p bzw. t_q stehen, verschieden, etwa $s_p < s_q$, so gilt $p < q$.
- (iii) Für $p = 1, \dots, k$ gilt $a_p = \max \{m_p^E, m_p^L\}$ mit $m_p^E = \max \{e_q \mid 1 \leq q < p, i_q = i_p\}$ und $m_p^L = \max \{e_q \mid 1 \leq q < p, j_q = j_p\}$, wobei $\max \emptyset := 0$.

In einem zulässigen Schedule $S = (t_1, \dots, t_k)$ sind wegen (ii) und $a_p \geq m_p^E$ nach (iii) für $p = 1, \dots, k$ alle Reihenfolge-Restriktionen der Datentabelle berücksichtigt:

- Sei t_p der zu einem Sprechwunsch des j -ten Lehrers durch das i -te Elternpaar gehörende Termin. Weiter repräsentiere t_q einen Sprechwunsch des j' -ten Lehrers durch dasselbe Elternpaar i . Es sei angenommen, dass t_p gemäß der Spalteneinteilung der Datentabelle vor t_q stattfinden muss. Dann steht t_p in S „links“ von t_q , d.h. es ist $p < q$. Ferner gilt $a_p < a_q$.
- Nach (iii) gilt $a_p \geq m_p^L$ für alle p . Weiter „links“ stehende Termine finden folglich früher als weiter „rechts“ stehende Termine desselben Lehrers mit verschiedenen Elternpaaren statt.

Man beachte, dass die Forderung der letzteren Eigenschaft die Sprechzeitenkriterien (1) und/oder (2) verschlechtern kann. Ein Lehrer darf nämlich weiter „rechts“ stehende Termine zeitlich nicht vorziehen, selbst wenn die Dauer des Termins ein Einfügen in eine Lücke seines Terminkalenders erlauben würde, ohne anderweitige Reihenfolge-Prioritäten des entsprechenden Elternpaars zu verletzen. Die Berechnung von a_p nach (iii) lässt sich jedoch einfach implementieren und ausführen. Zudem lässt die im Folgenden beschriebene lokale Suche den Austausch von Terminpaaren zu. Dieses Manko wird daher durch das Suchverfahren behoben.

Der **Lösungsraum** ist die Menge $V \subseteq T^k$ der zulässigen Schedules. Man kann einen zulässigen Start-Schedule für lokale Suche wie folgt zufällig bestimmen: Man permutiere zufällig die Sprechtermine, die innerhalb einer Zeile und einer Spalte der Datentabelle stehen. Dann baue man $S = (t_1, \dots, t_k)$ schrittweise auf. Dazu würfle man für jedes $p \in \{1, \dots, k\}$ sukzessive einen Zeilenindex i und entnehme den von links her gesehen ersten, bis dato noch nicht gewählten Eintrag. Dabei trage man die Werte i_p , j_p und d_p ein. In einem zweiten p -Durchlauf ermittle man die Zeiten a_p und e_p gemäß der Formel in (iii). Das Erstellen von S wird bewusst in zwei p -Durchläufe aufgeteilt, da die Prozedur, die die Zeiten a_p und e_p einträgt,³ auch nach einem Übergang zu einem Nachbar-Schedule

³im Folgenden *Zeit-Einfüge-Prozedur* genannt

während lokaler Suche separat aufzurufen sein soll.

Die **Zielfunktion** $f : V \rightarrow \mathbb{R}$ einer Instanz setzt sich aus den beiden Bewertungen gemäß Kriterium (1) und Kriterium (2) zusammen. In dieser Arbeit wird

$$f(S) := 10^{10} \cdot g(S) + h(S)$$

$$\text{mit } g(S) := \max \{e_p \mid p = 1, \dots, k\}$$

$$\text{und } h(S) := \sum_{i=1}^n (\max \{e_p \mid p = 1, \dots, k, i_p = i\} - \min \{a_p \mid p = 1, \dots, k, i_p = i\})$$

gesetzt. Die Wahl des Faktors 10^{10} ist willkürlich und vielleicht auch größer angesetzt als nötig, um aus $f(S)$ die Teilwerte $g(S)$ und $h(S)$ zu rekonstruieren. Andererseits ist dieser Faktor aber auch bei größeren n und m als 20 und 15 noch ausreichend.

Zwei zulässige Schedules $S = (t_1, \dots, t_k)$ und $S' = (t'_1, \dots, t'_k)$ seien genau dann benachbart, wenn $p, q \in \{1, \dots, k\}$ existieren mit $1 \leq p < q \leq k$ und

$$t'_p = t_q, \quad t'_q = t_p, \quad t'_r = t_r \quad \text{für alle } r \in \{1, \dots, k\} \setminus \{p, q\}.$$

Benachbarte zulässige Schedules gehen also durch Austausch von zwei Terminen auseinander hervor. Im Folgenden soll deshalb von der **Paar-Austausch-Nachbarschaft** gesprochen werden. Jedes $S \in V$ hat maximal $d := \binom{k}{2}$ Nachbarn. Um während lokaler Suche bei aktuellem zulässigem Schedule $S = (t_1, \dots, t_k) \in V$ zu testen, ob der Austausch zweier Termine t_p und t_q mit $1 \leq p < q \leq k$ zulässig ist, gehe man wie folgt vor: Man tausche t_p und t_q und prüfe nach, ob sowohl die Termine $t_q, t_{p+1}, t_{p+2}, \dots, t_{q-1}$ laut Datentabelle vor t_p stattfinden dürfen als auch die Termine $t_{p+1}, t_{p+2}, \dots, t_{q-1}$ nach t_q . Sind Reihenfolgeprioritäten verletzt, werden t_q und t_p wieder zurückgetauscht. Ansonsten wird ein Update der Anfangs- und Endzeiten der Termine mittels der Zeit-Einfüge-Prozedur durchgeführt. Man erhält dann einen benachbarten zulässigen Schedule von S .

Analog zur vollen bzw. schnellen lokalen Suche im Lösungsraum $\{0, 1\}^n$ (Algorithmen 1.3, 1.4 in Abschnitt 1.1) lassen sich auch für den Lösungsraum V und die Paar-Austausch-Nachbarschaft entsprechende Verfahren formulieren. Durch lexikographische Anordnung der Paare (p, q) mit $1 \leq p < q \leq k$ ergibt sich eine kanonische Reihenfolge der d potentiellen Kandidaten für zulässige Nachbar-Schedules von S . Ersetzt man in Algorithmus 1.5 die n Nachbarn $y^1(x), y^2(x), \dots, y^n(x)$ von $x \in \{0, 1\}^n$ durch die d Nachbar-Kandidaten von S in der kanonischen Reihenfolge, so ist insbesondere zyklische schnelle lokale Suche möglich.

Es bleibt zu klären, wie man ein Analogon zur Mehrheitsbildung im $\{0, 1\}^n$ für zulässige Schedules formulieren könnte. Die Idee ist, Mehrheitsbildung durch die Mittelwertbildung der Anfangszeiten zu ersetzen:

DEFINITION A.2 (Mittelwertbildung)

Sei $L \geq 2$, und seien $S^\ell = (t_1^\ell, t_2^\ell, \dots, t_k^\ell)$ mit $t_p^\ell = (i_p^\ell, j_p^\ell, d_p^\ell, a_p^\ell, e_p^\ell)$ für $\ell = 1, \dots, L$ und $p = 1, \dots, k$ zulässige Schedules. Für jedes Paar $(i, j) \in \{1, \dots, n\} \times \{1, \dots, m\}$, das zu einem Terminwunsch des i -ten Elternpaares mit dem j -ten Lehrer gehört, sei $p(\ell, i, j)$ dasjenige Element p von $\{1, \dots, k\}$, für das $(i_p^\ell, j_p^\ell) = (i, j)$ gilt, d.h. $p(\ell, i, j)$ indiziere

den Termin von Paar i und Lehrer j im Schedule S^ℓ . Setze

$$s(i, j) := \sum_{\ell=1}^L a_{p(\ell, i, j)}^\ell$$

für jedes (i, j) und sortiere die k Paare (i, j) aufsteigend nach $s(i, j)$. Für $p = 1, \dots, k$ sei (i_p, j_p) das p -te Paar gemäß dieser Reihenfolge, d_p die Dauer des Gesprächs des i_p -ten Elternpaars und j_p -ten Lehrers und a_p sowie e_p die Anfangs- und Endzeiten, ermittelt durch die Zeit-Einfüge-Prozedur.

Dann heißt $S := (t_1, t_2, \dots, t_k)$ mit $t_p = (i_p, j_p, d_p, a_p, e_p)$ für $p = 1, \dots, k$ ein durch **Mittelwertbildung** aus S^1, S^2, \dots, S^L gewonnener Schedule.

BEMERKUNG A.3

Sei $L \geq 2$, und seien $S^\ell = (t_1^\ell, t_2^\ell, \dots, t_k^\ell)$ mit $t_p^\ell = (i_p^\ell, j_p^\ell, d_p^\ell, a_p^\ell, e_p^\ell)$ für $\ell = 1, \dots, L$ und $p = 1, \dots, k$ zulässige Schedules. Weiter sei $S = (t_1, t_2, \dots, t_k)$ mit $t_p = (i_p, j_p, d_p, a_p, e_p)$ für $p = 1, \dots, k$ ein durch Mittelwertbildung aus S^1, S^2, \dots, S^L gewonnener Schedule. Dann ist S zulässig.

BEWEIS. Es ist (ii) von Definition A.1 zu zeigen. Seien also $p, q \in \{1, \dots, k\}$ mit $i := i_p = i_q$ so, dass das i -te Elternpaar den j_p -ten Lehrer laut Datentabelle noch vor dem j_q -ten Lehrer sprechen soll. Da S^1, S^2, \dots, S^L nach Voraussetzung zulässig sind, gilt

$$p(\ell, i, j_p) < p(\ell, i, j_q) \quad \text{für } \ell = 1, \dots, L,$$

also

$$a_{p(\ell, i, j_p)}^\ell < e_{p(\ell, i, j_p)}^\ell \leq a_{p(\ell, i, j_q)}^\ell \quad \text{für } \ell = 1, \dots, L$$

und somit

$$s(i, j_p) = \sum_{\ell=1}^L a_{p(\ell, i, j_p)}^\ell < \sum_{\ell=1}^L a_{p(\ell, i, j_q)}^\ell = s(i, j_q).$$

Nach Konstruktion von S folgt $p < q$. □

BEISPIEL A.4

Seien $n = 3$, $m = 3$ und $k = 5$. Die Sprechwünsche der drei Elternpaare seien durch folgende Tabelle festgelegt:

Elternpaar	Sprechwünsche
1	(A1, C2)
2	(A3, B1)
3	C3

Weiter seien die zulässigen Schedules $S^\ell = (t_1^\ell, t_2^\ell, \dots, t_5^\ell)$ für $\ell = 1, 2, 3$ mit

$$\begin{array}{lll}
t_1^1 = (2, 1, 3, 0, 3) & t_1^2 = (1, 3, 2, 0, 2) & t_1^3 = (2, 2, 1, 0, 1) \\
t_2^1 = (3, 3, 3, 0, 3) & t_2^2 = (2, 2, 1, 0, 1) & t_2^3 = (3, 3, 3, 0, 3) \\
t_3^1 = (1, 3, 2, 3, 5) & t_3^2 = (1, 1, 1, 2, 3) & t_3^3 = (2, 1, 3, 1, 4) \\
t_4^1 = (2, 2, 1, 3, 4) & t_4^2 = (3, 3, 3, 2, 5) & t_4^3 = (1, 1, 1, 4, 5) \\
t_5^1 = (1, 1, 1, 5, 6) & t_5^2 = (2, 1, 3, 3, 6) & t_5^3 = (1, 3, 2, 5, 7)
\end{array}$$

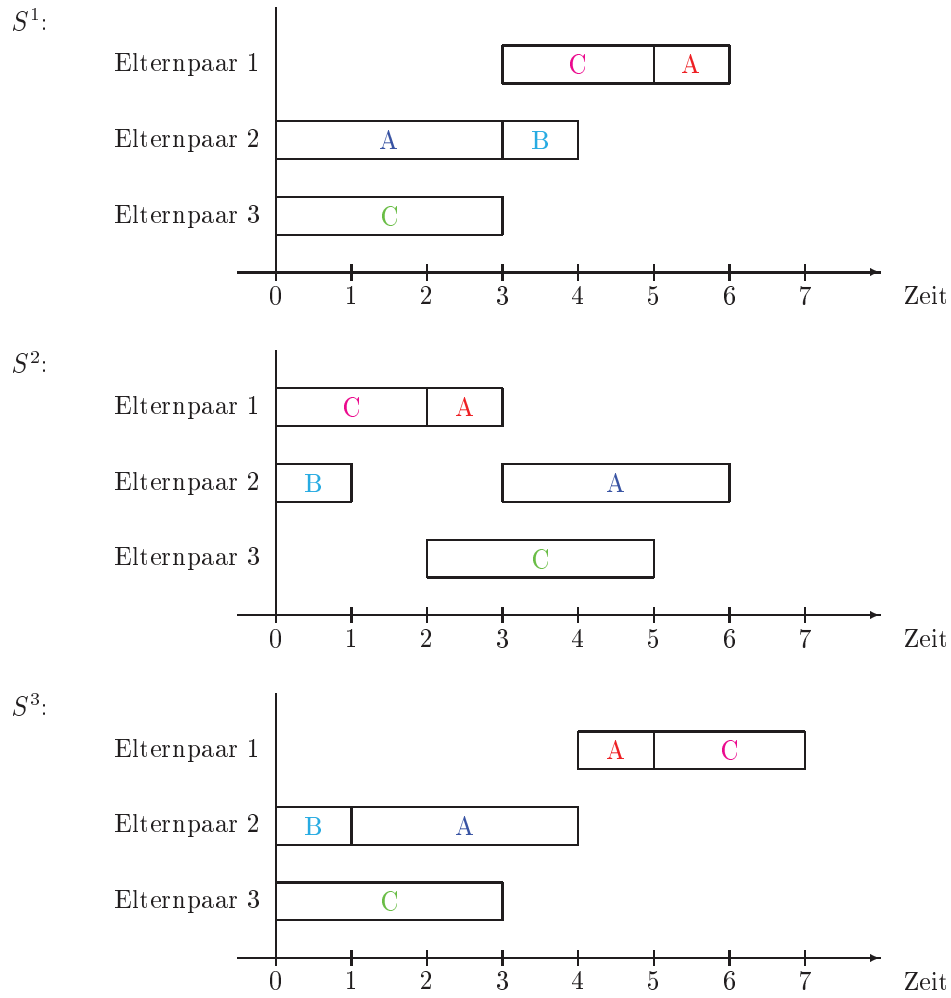


Abbildung A.2: GANTT-Diagramm-Darstellung der Schedules S^1 , S^2 , S^3 aus Beispiel A.4

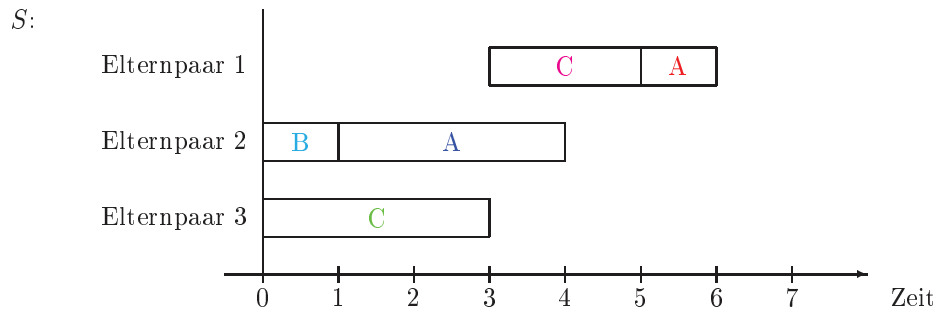
gegeben. S^1 , S^2 und S^3 sind zusätzlich in Abbildung A.2 als sogenannte GANTT-Diagramme dargestellt. Gehe nun $S = (t_1, t_2, \dots, t_5)$ durch Mittelwertbildung aus S^1, S^2, S^3 hervor. Dann enthalten die fünf Termine t_1, t_2, \dots, t_5 von S die fünf Sprechwünsche der Datentabelle, aufsteigend sortiert nach der Summe der Anfangszeiten der entsprechenden Termine in S^1 , S^2 und S^3 . Es gilt:

$$\begin{aligned}
 s(1, 1) &= 5 + 2 + 4 = 11 \\
 s(1, 3) &= 3 + 0 + 5 = 8 \\
 s(2, 1) &= 0 + 3 + 1 = 4 \\
 s(2, 2) &= 3 + 0 + 0 = 3 \\
 s(3, 3) &= 0 + 2 + 0 = 2
 \end{aligned}$$

Man erhält somit

$$\begin{aligned}
 t_1 &= (3, 3, 3, 0, 3), \\
 t_2 &= (2, 2, 1, 0, 1), \\
 t_3 &= (2, 1, 3, 1, 4), \\
 t_4 &= (1, 3, 2, 3, 5), \\
 t_5 &= (1, 1, 1, 5, 6).
 \end{aligned}$$

Abbildung A.3 zeigt S als GANTT-Diagramm.

Abbildung A.3: GANTT-Diagramm-Darstellung des Schedules S aus Beispiel A.4

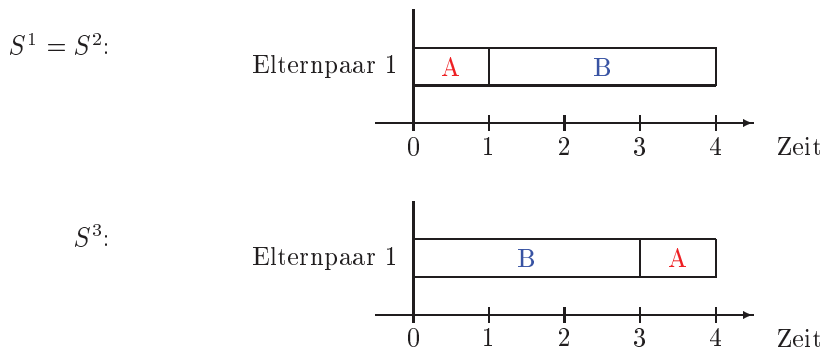
In gewissem Sinne wird durch die Mittelwertbildung gemäß Definition A.2 ein „Konsens“ der Ausgangs-Schedules S^1, S^2, \dots, S^L in S übernommen. Beginnt nämlich ein durch (i, j) gegebener Sprechwunsch in *jedem* der L zulässigen Schedules S^1, \dots, S^L vor einem durch (i', j') festgelegten Sprechwunsch, so gilt dies auch in S . Speziell in Beispiel A.4 und den angegebenen zulässigen Schedules S^1, S^2, S^3 und S bleiben auch alle *mehrheitlichen* zeitlichen Vorrangigkeiten erhalten: Beginnt der Termin des i -ten Elternpaares und j -ten Lehrers in *mindestens* zwei der drei Schedules S^1, S^2, S^3 vor dem Termin des i' -ten Elternpaares mit dem j' -ten Lehrer ($1 \leq i, j, i', j' \leq 3$), so passiert dies auch in S . Man prüfe dies für jeden Termin nach. Im Allgemeinen reicht die mehrheitliche zeitliche Vorrangigkeit jedoch nicht aus. Betrachte etwa folgendes denkbar einfache

BEISPIEL A.5

Gegeben seien $n = 1$, $m = 2$, $k = 2$ und folgende Sprechwunsch-Tabelle:

Elternpaar	Sprechwünsche
1	(A1, B3)

Weiter seien $L = 3$ und S^1, S^2, S^3 durch Abbildung A.4 festgelegt. Dann ergeben sich als Summen der Anfangszeiten der beiden Sprechwünsche $s(1, 1) = 0 + 0 + 3 = 3$ und $s(1, 2) = 1 + 1 + 0 = 2$. Für den durch Mittelwertbildung aus S^1, S^2, S^3 entstehenden Schedule S folgt also $S = S^3$. Obwohl Lehrer A in S^1 und S^2 vor Lehrer B gesprochen wird, ist die Reihenfolge in S umgekehrt.

Abbildung A.4: GANTT-Diagramm-Darstellung der Schedules S^1, S^2, S^3 aus Beispiel A.5

Als grobe Faustregel für beliebige Instanzen gilt dennoch, dass Termine, die in S^1, S^2, \dots, S^L mehrheitlich vor anderen Terminen beginnen, typischerweise auch in S zu einem frü-

heren Anfangszeitpunkt stattfinden. Dies rechtfertigt die Bezeichnung „Konsens“ aus S^1, S^2, \dots, S^L für S . Geht der Start-Schedule eines Laufs lokaler Suche durch Mittelwertbildung aus zulässigen Schedules hervor, soll von einem **Konsenslauf** gesprochen werden. Läufe lokaler Suche, die in zufällig bestimmten zulässigen Schedules starten (vgl. S. 106), heißen **zufällige Läufe**. In den folgenden Abschnitten werden Konsensläufe lokaler Suche mit zufälligen Läufen verglichen. Dazu werden analoge Fragestellungen wie in den Kapiteln II bis IV zur Mehrheitsbildung behandelt.

A 1 Ergebnisse bei „Whizzkids ’97“

In diesem Abschnitt werden für das Benchmark „Whizzkids ’97“ (vgl. S. 104) Ergebnisse lokaler Suche mit Mittelwertbildung vorgestellt. Die Mittelwertbildung wurde analog zur Mehrheitsbildung beim Max-SAT-Benchmark „Hanoi 5“ (vgl. Abschnitte 2.1.1 und 4.2.1) sowohl einstufig als auch wiederholt im Rahmen eines Genetischen Algorithmus untersucht.

A 1.1 Häufigkeitsverteilungen lokaler Minima aus zufälligen Läufen und Konsensläufen

In einem ersten Versuch zu „Whizzkids ’97“ wurde getestet, zu welchen Zielfunktionswerten Konsensläufe im Vergleich zu zufälligen Läufen führen. Aus Laufzeitgründen wurde auf den Einsatz voller lokaler Suche verzichtet. Stattdessen wurden die wiederholte schnelle lokale Suche sowie die probabilistisch wiederholte zyklische schnelle lokale Suche verwendet. Von beiden Varianten wurden 4000 Läufe durchgeführt, nämlich 1000 Blöcke zu je drei zufälligen Läufen und einem Konsenslauf. Die 4000 gefundenen lokal minimalen zulässigen Schedules waren beide Male paarweise verschieden. Pro Variante der wiederholten lokalen Suche traten die gefundenen f -Werte im wesentlichen nur je einmal auf. In Einzelfällen kam ein Wert doppelt vor, jedoch nicht öfter als zweimal.

Die Säulendiagramme in den Abbildungen A.5 und A.6 zeigen die Häufigkeitsverteilungen der aufgetretenen g -Werte, d.h. Zielfunktionswerte nach Kriterium (1), der 4000 lokal minimalen Lösungen. Aus optischen Gründen sind in jeder Säule die Häufigkeiten von acht aufeinander folgenden g -Werten zusammengefasst: In einer Säule werden alle lokal minimalen Lösungen S gezählt, deren Werte $g(S)$ in einem Bereich $8z \leq g(S) \leq 8z + 7$ für ein $z \in \{0, 1, 2, \dots\}$ liegen. Schwarze Säulen beziehen sich auf zufällige Läufe, rote Säulen auf Konsensläufe.

Der beste in einem zufälligen Lauf wiederholter schneller lokaler Suche gefundene Schedule S hatte Zielfunktionswert $g(S) = 547$, der schlechteste 852. In den Konsensläufen wurden lokal minimale Lösungen mit g -Werten zwischen 557 und 751 erreicht. Bei der probabilistisch wiederholten zyklischen schnellen lokalen Suche traten in den zufälligen Läufen g -Werte zwischen 551 und 813 und in den Konsensläufen zwischen 544 und 754 auf. In Tabelle A.1 sind ferner die g -Werte der Start-Schedules („ \bar{g}_{start} “), die g -Werte der ge-

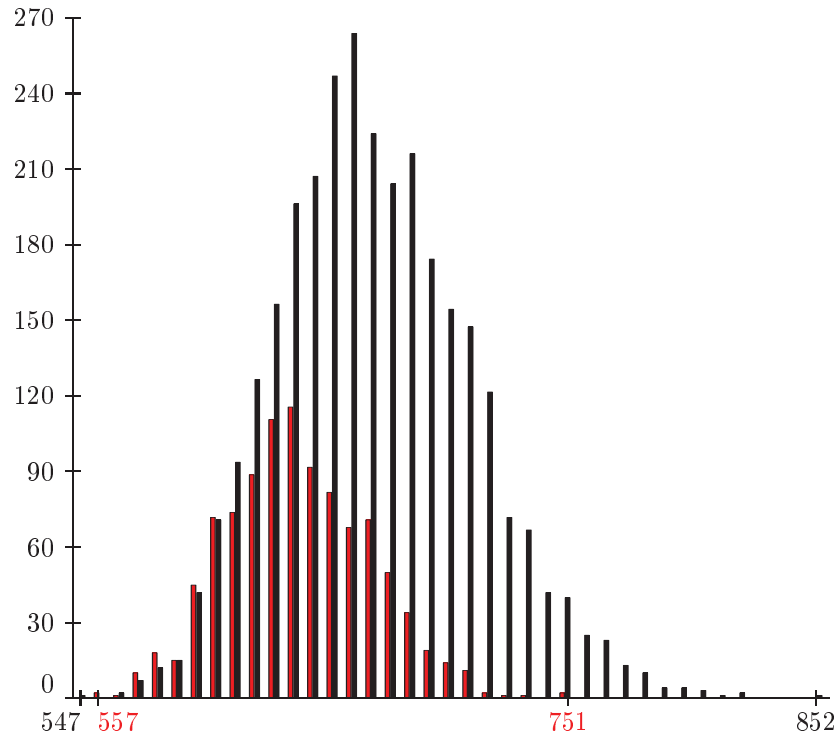


Abbildung A.5: Häufigkeitsverteilung der gefundenen lokalen Minima bei wiederholter schneller lokaler Suche bei „Whizzkids '97“

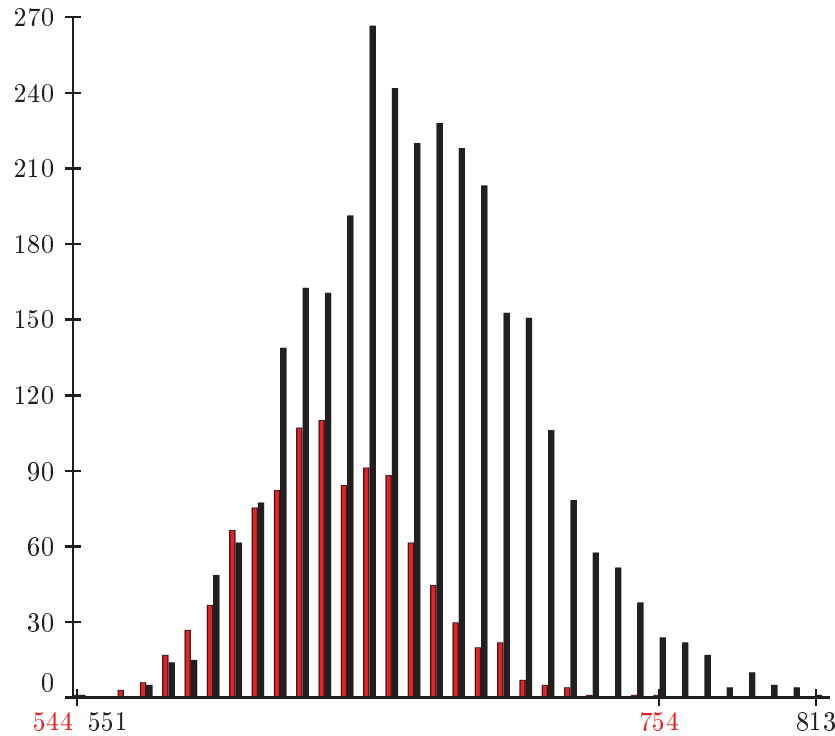


Abbildung A.6: Häufigkeitsverteilung der gefundenen lokalen Minima bei probabilistisch wiederholter zyklischer schneller lokaler Suche bei „Whizzkids '97“

gefundenen lokal minimalen Schedules („ \overline{g} “), die Schrittzahlen („ \overline{S} “) und die Fragenzahlen („ \overline{F} “) eingetragen. All diese Daten wurden jeweils über die 3000 zufälligen Läufe (Index

„zf“) bzw. 1000 Konsensläufe (Index „Ko“) lokaler Suche gemittelt. „S“ steht für die wiederholte schnelle lokale Suche, „pZS“ für die probabilistisch wiederholte zyklische schnelle lokale Suche.

	$\bar{g}_{\text{start,zf}}$	$\bar{g}_{\text{start,Ko}}$	\bar{g}_{zf}	\bar{g}_{Ko}	\bar{S}_{zf}	\bar{S}_{Ko}	\bar{F}_{zf}	\bar{F}_{Ko}
S	1134.7	976.0	668.7	637.7	215.8	176.1	16490.6	17159.0
pZS	1138.9	977.5	669.6	638.5	212.0	174.0	13487.4	13985.7

Tabelle A.1: *Ergebnisse aus zufälligen Läufen und Konsensläufen lokaler Suche verschiedener Varianten bei „Whizzkids ’97“*

Die Zahlen im Vergleich zeigen, dass Konsensläufe typischerweise bessere Schedules als zufällige Läufe liefern. Ferner sind ihre zurückgelegten Wege vom Start- zum Ziel-Schedule im Mittel kürzer. Dies lässt sich mit den besseren g -Werten der durch Mittelwertbildung gewonnenen Start-Schedules im Vergleich zu den zufällig erzeugten Schedules erklären. Konsensläufe stellen jedoch durchschnittlich etwas mehr Fragen an Nachbarn als zufällige Läufe. Bei Gegenüberstellung von schneller und zyklischer schneller lokaler Suche ergeben sich hinsichtlich der Güte der lokal minimalen Lösungen keine nennenswerten Unterschiede. Die Läufe der probabilistisch wiederholten zyklischen schnellen lokalen Suche weisen aber im Mittel weniger Fragen an benachbarte zulässige Schedules auf.

A 1.2 Wiederholte Mittelwertbildung als Genetischer Algorithmus

Der Genetische Algorithmus 4.2, Seite 93, für wiederholte Mehrheitsbildung lässt sich nahezu wörtlich auf wiederholte Mittelwertbildung übertragen:

ALGORITHMUS A.6

Seien $L \geq 2$ die Gruppengröße, $\varrho \geq 1$ die Poolgröße und $S \geq 1$ die maximale Stufe.

Schritt 1: Setze $t := 0$.

Mache ϱ zufällige Läufe lokaler Suche. Seien $x^0, x^1, \dots, x^{\varrho-1}$ die lokal minimalen zulässigen Schedules. Setze $\mathcal{P} := (x^0, x^1, \dots, x^{\varrho-1})$.

Schritt 2: Wähle zufällig eine Permutation $\pi \in S_{\varrho}$. Permutiere $\mathcal{P} = (x^0, x^1, \dots, x^{\varrho-1})$ zu $\mathcal{P} := (x^{\pi(0)}, x^{\pi(1)}, \dots, x^{\pi(\varrho-1)})$.

Schritt 3: Bilde aus der Menge $\{0, 1, \dots, \varrho-1\}$ „Gruppen“ $G_0, G_1, \dots, G_{\varrho-1}$ der Größe L gemäß Bemerkung 4.1.

Schritt 4: Für $i = 0, \dots, \varrho-1$ sei y^i ein zulässiger Schedule, der durch Mittelwertbildung aus den durch G_i indizierten zulässigen Schedules hervorgeht. Mache ϱ Konsensläufe lokaler Suche mit Start-Schedules $y^0, y^1, \dots, y^{\varrho-1}$. Seien $z^0, z^1, \dots, z^{\varrho-1}$ die lokal minimalen zulässigen Schedules. Setze $\mathcal{P} := (z^0, z^1, \dots, z^{\varrho-1})$.

Schritt 5: Falls $t = S$, stopp. Gib den insgesamt besten gefundenen zulässigen Schedule und seinen Zielfunktionswert aus. Sonst setze $t := t + 1$ und gehe zu Schritt 2.

Im Folgenden werden Ergebnisse von Algorithmus A.6 für das Benchmark „Whizzkids ’97“ vorgestellt. Seien $L = 2$, $\varrho = 4$ und $S = 99$. Als Variante lokaler Suche wurde die probabilistisch wiederholte zyklische schnelle lokale Suche ohne Sidesteps eingesetzt. Algorithmus A.6 wurde zweimal unabhängig gestartet. Daraus ergaben sich 800 Läufe lokaler Suche. Die Ergebnisse sind Tabelle A.2 zu entnehmen.

t	$(g, h)_t^*$	\bar{g}_t	\bar{S}_t	\bar{F}_t
0	(585,6463)	663.3	215.7	13042.1
1	(593,6225)	625.5	170.6	14067.1
2	(600,6481)	622.4	148.6	11518.8
3	(583,6167)	608.8	137.8	10875.4
4	(565,5975)	603.2	107.8	8880.1
5	(566,6026)	600.1	105.3	9155.1
6	(573,5948)	603.0	91.9	8681.8
7	(566,5821)	595.8	91.8	9379.4
8	(578,5892)	602.0	89.8	8868.8
9	(568,5742)	593.6	78.3	8702.3
10	(552,5655)	588.8	91.6	8672.2
\vdots	\vdots	\vdots	\vdots	\vdots
58	(528,5702)	550.4	46.1	6758.9
\vdots	\vdots	\vdots	\vdots	\vdots
99	(531,5581)	551.4	38.4	5636.8

Tabelle A.2: *Ergebnisse des Genetischen Algorithmus mit $S = 99$, $\varrho = 4$, $L = 2$ und probabilistisch wiederholter zyklischer schneller lokaler Suche (**ohne Sidesteps**) bei „Whizzkids ’97“*

Mit $t \in \{0, 1, \dots, 99\}$ wird die Generation der aktuellen Population bezeichnet. $(g, h)_t^*$ ist das Paar $(g(x), h(x))$ eines zulässigen Schedules x mit kleinstem f -Wert in Generation t . Der beste gefundene zulässige Schedule x^* aus den 800 Läufen trat in Generation 58 mit Zielfunktionswerten $(g(x^*), h(x^*)) = (528, 5702)$ auf. \bar{g}_t bezeichnet den Durchschnittswert der g -Werte der vier zulässigen Schedules in Generation t . Mit wachsendem t ist bis zu einer bestimmten Generation (hier 5) eine ständige Abnahme von \bar{g}_t zu beobachten. Danach nimmt \bar{g}_t nur noch „im Groben“ weiter ab. Zwischendurch wird immer wieder ein Ansteigen des Durchschnittswerts ersichtlich. Bei den durchschnittlichen Schrittzahlen \bar{S}_t der vier Läufe lokaler Suche, welche zu den zulässigen Schedules in Generation t führen, zeigt sich ein ähnlicher Effekt. $\bar{S}_0, \bar{S}_1, \dots, \bar{S}_9$ fallen monoton in t . Für $t > 9$ nimmt \bar{S}_t insgesamt weiter ab, jedoch nicht in jeder Generation. Die durchschnittlich benötigten Fragenzahlen \bar{F}_t werden schließlich von Generation 0 zu Generation 99 deutlich geringer. Doch auch dieses Abfallen findet nicht „kontinuierlich“ statt.

Weitere Versuche zu Algorithmus A.6 zeigten, dass die beschriebenen Effekte typisch sind. Man beachte, dass in dem Genetischen Algorithmus *keine* Mutationen in Form von zufälligen Vertauschungen von Terminen eines zulässigen Schedules stattfinden. Ferner wird die Population jeder Generation vollständig durch ihre Nachkommen ersetzt. Insbesondere wird auf *Elitismus* verzichtet, d.h. gegebenenfalls werden gute zulässige Schedules rigoros

durch schlechtere ausgetauscht. Die Fitness der Individuen, also die Zielfunktionswerte der zulässigen Schedules, spielen ferner nur während lokaler Suche eine Rolle. Sie werden weder bei der Elternwahl noch bei der Selektion berücksichtigt. Trotz dieser erheblichen Vereinfachungen gegenüber einem „traditionellen“ Genetischen Algorithmus sind die Individuen höherer Generationen im Durchschnitt besser als in früheren Generationen. Dies zeigt, dass die Mittelwertbildung zur Rekombination gut geeignet ist.

In einem weiteren Versuch wurden Sidesteps erlaubt. Eine Folge von Sidesteps wurde nach 1000 Übergängen zu benachbarten zulässigen Schedules mit gleichem f -Wert abgebrochen. Es wurde also die in Abschnitt 1.1.5 beschriebene Schrankenmethode verwendet. Alle weiteren Größen und Parameter waren wie oben. Die Ergebnisse dieses zweiten Versuchs sind in Tabelle A.3 dargestellt. Mit \overline{S}_t werden hier die durchschnittlichen Übergänge zu echt besseren zulässigen Schedules während der Läufe lokaler Suche bezeichnet, die die Schedules von Generation t liefern. $\overline{\text{side}}_t$ ist die durchschnittliche Zahl der aufgetretenen Sidesteps während eines Laufs zu Generation t .

t	$(g, h)_t^*$	\overline{g}_t	\overline{S}_t	$\overline{\text{side}}_t$	\overline{F}_t
0	(561,5491)	618.6	352.3	8568.8	114149.8
1	(563,5986)	602.6	256.3	9267.8	110583.3
2	(534,6049)	584.6	228.2	8235.3	98337.3
3	(534,5791)	564.3	210.8	8375.3	94177.6
4	(530,5539)	559.9	158.6	5361.0	60769.8
5	(534,5652)	554.5	154.2	5004.2	55193.2
6	(529,5398)	552.0	135.9	5250.5	54849.4
7	(523,5538)	547.6	133.6	6383.4	63312.3
8	(529,5386)	548.3	124.6	6780.1	63387.6
9	(526,5632)	547.3	105.8	5998.3	55993.7
10	(524,5447)	546.8	100.9	4648.2	44124.4
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
26	(512,5150)	539.3	51.6	5019.7	38566.0
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
99	(512,5272)	530.3	24.4	3683.4	25558.0

Tabelle A.3: *Ergebnisse des Genetischen Algorithmus mit $S = 99$, $\varrho = 4$, $L = 2$ und probabilistisch wiederholter zyklischer schneller lokaler Suche (mit Sidesteps bzgl. f) bei „Whizzkids ’97“*

Die gefundenen Zielfunktionswert-Paare waren insgesamt deutlich besser als ohne Erlaubnis von Sidesteps. \overline{g}_t , \overline{S}_t , $\overline{\text{side}}_t$ und \overline{F}_t nahmen tendenziell von den frühen zu den späten Generationen hin ab. Der beste gefundene zulässige Schedule gehörte zur Population in Generation 26 und hatte f -Wert 512 sowie g -Wert 5150.

In einem abschließenden, dritten Versuch wurden wiederum Sidesteps zugelassen. Diese bezogen sich jedoch auf das erste Optimierungskriterium, d.h. die Zielfunktion g . Es waren also auch Übergänge zu benachbarten zulässigen Schedules mit gleichem g -Wert, aber schlechterem h -Wert erlaubt. Zwischen zwei g -Wert-Verbesserungen waren bis zu 1 000 000 Sidesteps bzgl. g möglich. Ferner wurde die maximale Generationenzahl S von 99

auf 49 herunter gesetzt. Dafür wurde die Populationsgröße ϱ von 2 auf 16 vergrößert. Insgesamt wurden wieder 800 Läufe wiederholter lokaler Suche gemacht. Die Ergebnisse sind in Tabelle A.4 dargestellt.

t	$(g, h)_t^*$	\bar{g}_t	\bar{S}_t	$\overline{\text{side}}_t$	\bar{F}_t
0	(490,6073)	508.4	151.3	1 799 874.2	5 111 458.7
1	(486,6240)	498.5	91.6	1 348 705.5	3 807 548.8
2	(483,6151)	493.1	80.8	1 299 025.6	3 700 123.9
3	(477,6153)	491.4	74.8	1 316 281.1	3 727 705.9
4	(474,5935)	486.1	80.4	1 304 112.3	3 738 139.2
5	(474,6036)	482.1	70.8	1 557 893.6	4 524 532.3
6	(474,6229)	481.9	72.8	1 478 937.6	4 292 183.9
7	(472,5814)	482.2	70.6	1 335 319.8	3 870 850.0
8	(474,5887)	481.1	72.2	1 627 938.4	4 710 199.6
9	(472,5888)	482.1	74.8	1 372 314.3	3 977 467.4
10	(472,5864)	482.3	63.5	1 307 809.2	3 783 754.4
11	(469,5920)	479.6	71.7	1 588 301.4	4 622 796.8
12	(476,5820)	481.8	70.9	1 452 227.3	4 192 524.8
13	(469,5884)	479.3	66.6	1 485 982.3	4 316 001.2
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
22	(469,5777)	477.0	65.4	1 327 745.0	3 862 726.8
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
49	(473,5880)	479.4	72.3	1 583 606.3	4 591 806.9

Tabelle A.4: *Ergebnisse des Genetischen Algorithmus mit $S = 49$, $\varrho = 16$, $L = 2$ und probabilistisch wiederholter zyklischer schneller lokaler Suche (**mit Sidesteps bzgl. g**) bei „Whizzkids ’97“*

Es fällt sofort auf, dass die Zielfunktionswerte erheblich besser sind als im vorangegangenen Versuch. Der beste zulässige Schedule hatte g -Wert 469 und h -Wert 5777 und stammte aus Generation 22. Der g -Wert ist also global minimal. Die Durchschnittswerte \bar{g}_t und \bar{S}_t nehmen wie schon in den vorangegangenen Versuchen tendenziell von den frühen zu den späten Generationen t hin ab. Die Werte $\overline{\text{side}}_t$ und \bar{F}_t werden dagegen nur von Generation 0 zu Generation 1 deutlich kleiner. Danach ist kein weiteres Abfallen mehr zu beobachten. Die hohen Sidestep- und Fragenzahlen zeigen, dass die lokale Suche massiv von der großen Zahl erlaubter aufeinanderfolgender Sidesteps Gebrauch macht.

A 2 Ergebnisse bei zufällig erzeugten Varianten von „Whizzkids ’97“

Um ein- und mehrstufige Mittelwertbildung in größerer Allgemeinheit zu testen, werden bestimmte Varianten des Scheduling-Problems „Whizzkids ’97“ behandelt. Im Folgenden wird beschrieben, wie sich Instanzen zufällig erzeugen lassen.

Sei $n \geq 1$ die Anzahl der Elternpaare. In Anlehnung an das Verhältnis zwischen der Eltern- und Lehrerzahl bei der Instanz „Whizzkids ’97“ wird als Anzahl an Lehrern $m := \lceil \frac{3}{4} n \rceil$ gesetzt. Weiter sei $\tilde{k} := \lceil \frac{nm}{2} \rceil$ als vorläufige Anzahl an Sprechterminen festgehalten. Im Durchschnitt soll also jedes Elternpaar etwa die Hälfte aller Lehrer sprechen. Dann wird für jedes Elternpaar $i \in \{1, \dots, n\}$ die endgültige Zahl k_i an Terminen festgelegt. Dazu wird zufällig und gleichverteilt eine ganze Zahl k_i aus dem Intervall $\left[\left\lceil \frac{\tilde{k}}{2n} \right\rceil, \left\lceil \frac{3\tilde{k}}{2n} \right\rceil \right]$ gewählt. Als Gesamtzahl aller Sprechwünsche ergibt sich $k := k_1 + k_2 + \dots + k_n$.

Zu den Problemparametern n , m und k wird nun eine Instanz als Tabelle analog zu Abbildung A.1 erstellt. Wähle dazu für jedes Elternpaar $i \in \{1, \dots, n\}$ paarweise verschiedene Zahlen $j_1^i, j_2^i, \dots, j_{k_i}^i \in \{1, \dots, m\}$, welche die vom i -ten Paar zu sprechenden Lehrer indizieren. Die Wahl erfolge zufällig und gleichverteilt auf der Menge aller k_i -elementigen Teilmengen von $\{1, \dots, m\}$. Sei $\pi \in S_{k_i}$ eine zufällig gewählte Permutation. Verteile die Zahlen $j_1^i, j_2^i, \dots, j_{k_i}^i$ in der Reihenfolge $\pi(j_1^i), \pi(j_2^i), \dots, \pi(j_{k_i}^i)$ auf die Spalten in Zeile i . Dabei wird $\pi(j_1^i)$ in Spalte 1 gelegt. Für $\ell > 1$ wird $\pi(j_\ell^i)$ mit Wahrscheinlichkeit $\frac{1}{3}$ in dieselbe Spalte wie $\pi(j_{\ell-1}^i)$ gelegt und mit Wahrscheinlichkeit $\frac{2}{3}$ in die nächste Spalte rechts. Ist \bar{a} die erwartete Anzahl an Einträgen einer bestimmten Spalte, so gilt

$$\bar{a} = 1 + \frac{1}{3} \cdot \bar{a} + \frac{2}{3} \cdot 0,$$

also $\bar{a} = \frac{3}{2}$. Die durchschnittlich $\frac{3}{2}$ Lehrer dürfen wie bei „Whizzkids ’97“ in beliebiger Reihenfolge von Elternpaar i gesprochen werden. Schließlich sei die Dauer d_j des j -ten Termins, $j = 1, \dots, k$, zufällig und gleichverteilt aus der Menge $\{1, 2, \dots, \lceil \frac{k}{3} \rceil\}$ gewählt. Damit ist eine zufällige Instanz der in dieser Arbeit behandelten Klasse von Scheduling-Problemen vollständig festgelegt.

A 2.1 Häufigkeitsverteilungen lokaler Minima aus zufälligen Läufen und Konsensläufen

Analog zu den Betrachtungen bei „Whizzkids ’97“ aus Abschnitt A 1.1 werden im Folgenden Häufigkeitsverteilungen lokaler Minima aus lokaler Suche bei zufällig erzeugten Instanzen vorgestellt. Für jedes $n \in \{1, 2, \dots, 40\}$ wurden 10 000 Instanzen zufällig gewählt. Bei jeder Instanz wurden vier Läufe probabilistisch wiederholter zyklischer schneller lokaler Suche gemacht, nämlich drei zufällige Läufe und ein anschließender Konsenslauf. Der Start-Schedule des Konsenslaufes entstand durch Mittelwertbildung aus den drei lokal minimalen Schedules der vorangegangenen zufälligen Läufe. Die lokalen Minima wurden nur nach dem ersten Optimalitätskriterium, d.h. in ihrem g -Wert, unterschieden.

Bei $n = 1$ erreichten alle 40 000 Läufe g -Wert 1. Dieses Ergebnis ist aufgrund der Größen $(n, m, k, d_1) = (1, 1, 1, 1)$ trivial. Bei $n = 2$ traten zwei verschiedene lokale Minima bzgl. g auf, nämlich 1 und 2. Beide lokalen Minima waren in der jeweils zu Grunde liegenden Instanz einziges lokales Minimum, so dass alle Läufe zu einer festen Instanz im gleichen Minimum terminierten. Auch dieses Ergebnis ist wegen des speziellen Konstruktionsschemas zufälliger Instanzen klar. Insbesondere waren die durchschnittlich aufgetretenen lokalen Minima aus den zufälligen Läufen sowohl bei $n = 1$ als auch bei $n = 2$ identisch

mit den Durchschnittswerten aus den Konsensläufen. Bei $n = 3$ ergaben sich erste Unterschiede. Hier lag der durchschnittliche g -Wert der Konsensläufe etwas unterhalb des durchschnittlichen g -Werts der zufälligen Läufe. In Abbildung A.7 sind die Häufigkeitsverteilungen der gefundenen lokalen Minima für $n = 10, 20, 30, 40$ skizziert. Mit schwarzer Farbe sind die Häufigkeiten der g -Werte aus den zufälligen Läufen dargestellt. Die roten Säulen kennzeichnen die Häufigkeiten der g -Werte aus Konsensläufen.

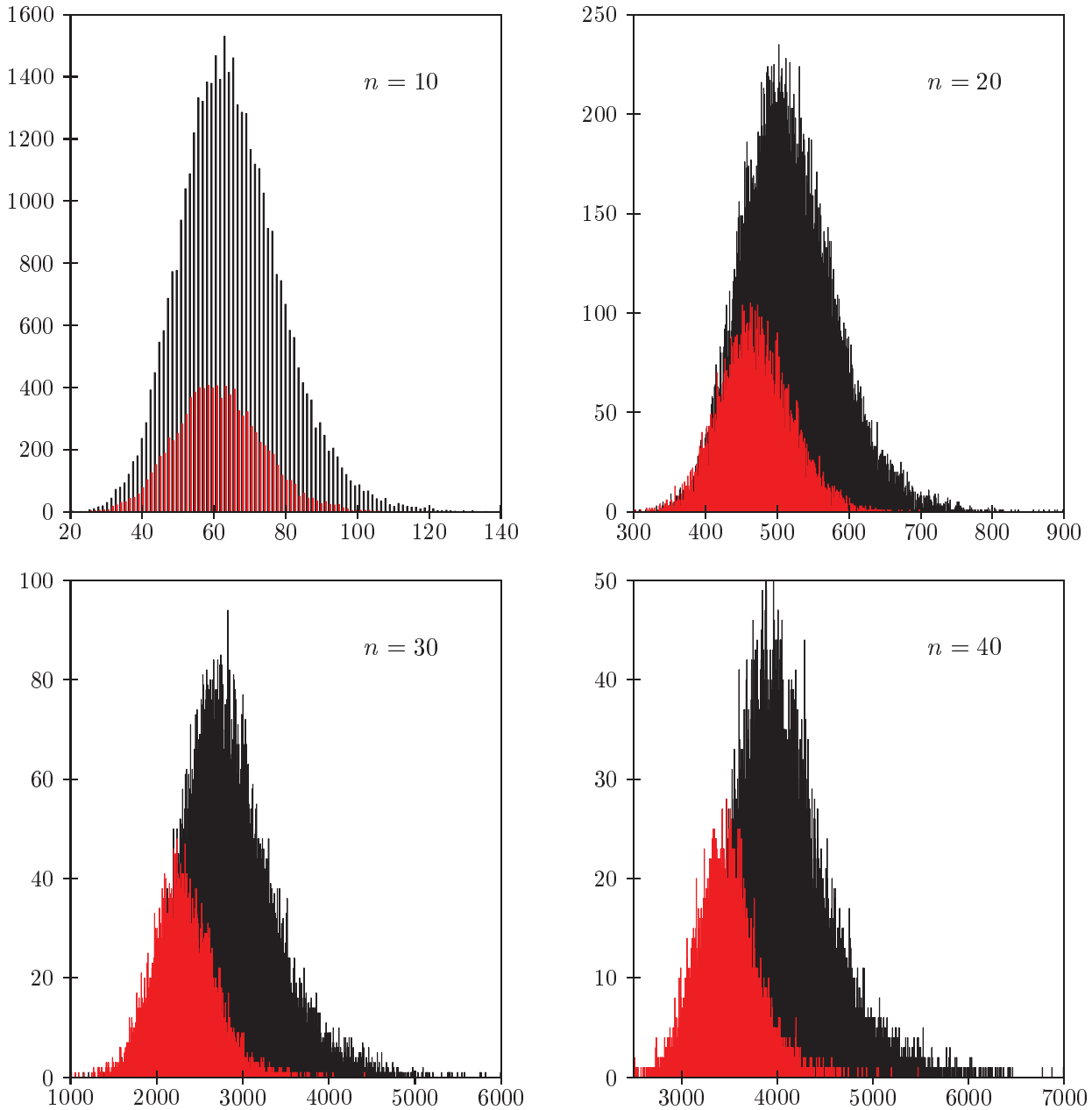


Abbildung A.7: **(Scheduling-Probleme)** Häufigkeitsverteilungen der gefundenen lokalen Minima unter probabilistisch wiederholter zyklischer schneller lokaler Suche bei zufällig gewählten Varianten von „Whizzkids '97“

Man erkennt, dass beide Verteilungen eine Glockengestalt aufweisen. Mit wachsendem n werden die Häufigkeiten fester g -Werte kleiner, d.h. es treten immer mehr verschiedene

g -Werte auf. Die wichtigste Beobachtung ist die, dass die rote Glocke mit wachsendem n nach links aus der schwarzen Glocke heraus wandert. Je stärker die rote Glocke aus der schwarzen heraustritt, desto höher ist die Wahrscheinlichkeit, dass der *beste* g -Wert aus einer bestimmten Zahl zufälliger Läufe und einem Konsenslauf aus letzterem stammt. Dieses Argument wird bei den Erfolgsquoten im nächsten Abschnitt eine Rolle spielen.⁴

A 2.2 Erfolgsquoten

Seien $n \in \{1, 2, 3, \dots, 60\}$ und $b \in \{1, 2, 3, 4, 5\}$ fest. Dann wurde folgender Versuch durchgeführt: Es wurden 10 000 unabhängige zufällige Instanzen zum Problemparameter n erzeugt. Pro Instanz wurden $b \cdot 4$ zufällige Läufe probabilistisch wiederholter zyklischer schneller lokaler Suche gemacht. Sei \min_{zf} der kleinste f -Wert der gefundenen $b \cdot 4$ lokal minimalen Lösungen. Weiter sei \min_{Ko} der kleinste f -Wert aus b Blöcken zu je drei zufälligen Läufen und einem anschließenden Konsenslauf. Die Startlösung des Konsenslaufs wurde jeweils durch Mittelwertbildung aus den lokal minimalen Lösungen der drei vorangegangenen zufälligen Läufe gewonnen. Während des Verfahrens wurde gezählt, wie oft $\min_{Ko} < \min_{zf}$ bzw. $\min_{Ko} = \min_{zf}$ bzw. $\min_{Ko} > \min_{zf}$ eintrat. Seien $\#\{\min_{Ko} < \min_{zf}\}$ bzw. $\#\{\min_{Ko} = \min_{zf}\}$ bzw. $\#\{\min_{Ko} > \min_{zf}\}$ die zugehörigen Anzahlen. Dann heißt

$$\frac{1 \cdot \#\{\min_{Ko} < \min_{zf}\} + \frac{1}{2} \#\{\min_{Ko} = \min_{zf}\}}{10\,000}$$

die (empirische) **Erfolgsquote** von Mittelwertbildung unter dem Problemparameter n , Gruppengröße 3, probabilistisch wiederholter zyklischer schneller lokaler Suche und Blockzahl b . Speziell für $b = 1$ ergibt sich ein Analogon zur Erfolgsquote von Mehrheitsbildung laut Definition 2.1. In Abbildung A.8 sind die Erfolgsquoten in Abhängigkeit von n skizziert. Die Graphik links zeigt die Erfolgsquoten unter $1 \leq n \leq 60$ und $b = 1$. Rechts sind die Erfolgsquoten unter $1 \leq n \leq 20$ und $1 \leq b \leq 5$ abgebildet.

Unter beliebigen Gruppengrößen $L \geq 2$ und Varianten S wiederholter lokaler Suche erfolgt die Begriffsbildung analog. Ferner wird der **Erfolg** von Mittelwertbildung unter n , L , S und b analog zum Erfolg von Mehrheitsbildung gemäß Definition 2.2 definiert.

VERMUTUNG A.7

Der Erfolg von Mittelwertbildung unter dem Problemparameter n , beliebiger Gruppengröße $L \geq 2$, beliebiger in Abschnitt 1.1 genannter Variante wiederholter lokaler Suche und beliebiger Blockzahl $b \geq 1$ konvergiert für $n \rightarrow \infty$ gegen 1.

Die Graphik in Abbildung A.8 rechts zeigt, dass die „Kurven“ der Erfolgsquoten umso „schneller“ wachsen, je größer b ist. Zudem lässt sich für jedes b ein **Misserfolgstal** erkennen: Für $n \in \{1, 2, \dots, 6\}$ und $b \in \{1, 2, \dots, 5\}$ sind die Erfolgsquoten höchstens 0.5. Alle Misserfolgstäler haben insbesondere dieselbe Länge: Unter jedem b ist 7 der kleinste Problemparameter n mit Erfolgsquote größer als 0.5. Die Misserfolgstäler lassen sich mit den Beobachtungen aus Abschnitt A 2.1 plausibilisieren. Je weiter links die rote Glocke

⁴vgl. auch Bemerkung 2 zu den Misserfolgstälern bei Dynamischen Optimierungsproblemen auf Seite 70.

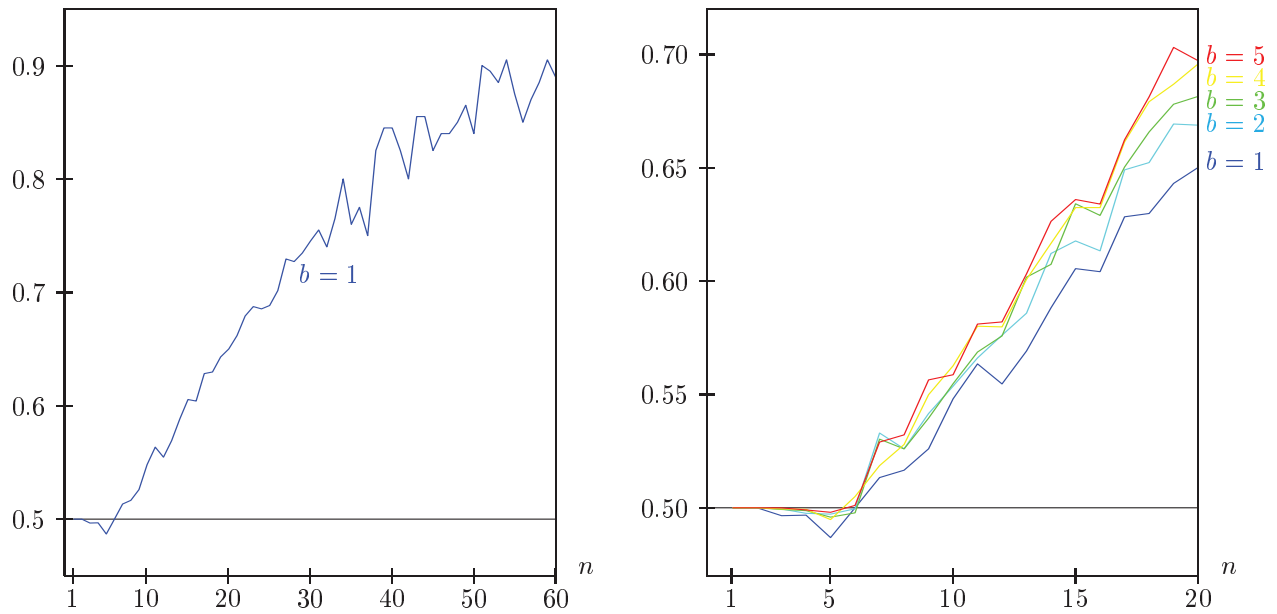


Abbildung A.8: **(Scheduling-Probleme)** Erfolgsquoten in Abhängigkeit von n unter Gruppengröße 3, probabilistisch wiederholter zyklischer schneller lokaler Suche und verschiedenen Blockzahlen b

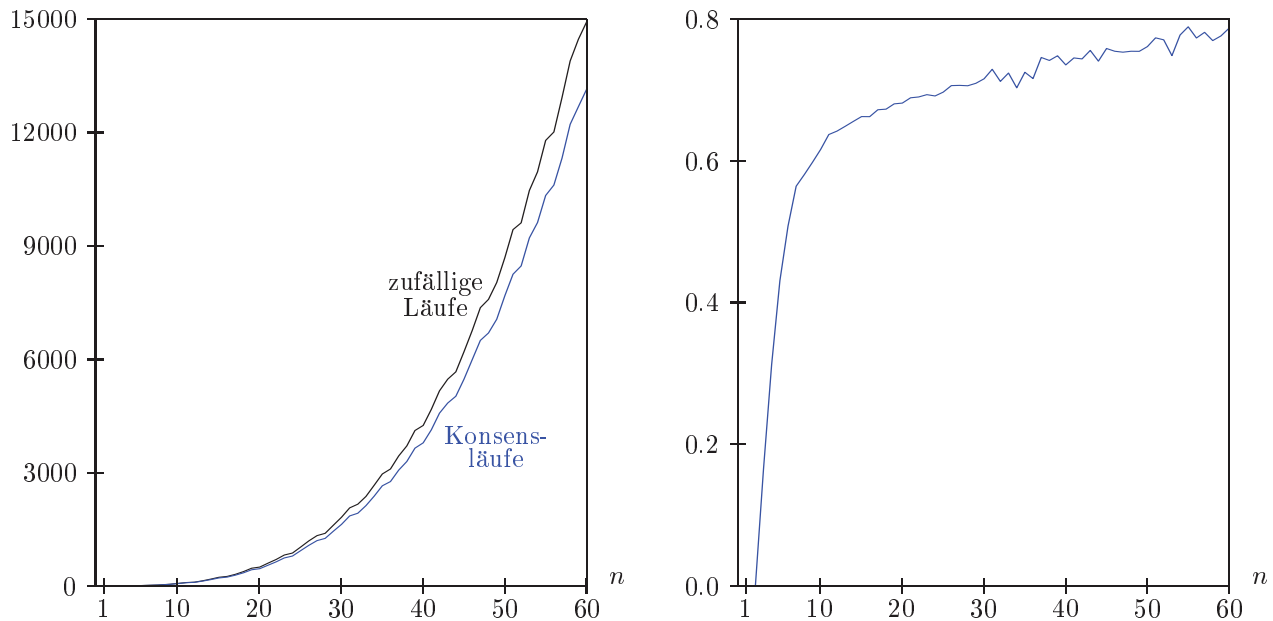


Abbildung A.9: **(Scheduling-Probleme)** Durchschnittlich erreichte g -Werte aus zufälligen Läufen und Konsensläufen (links) und Schrittzahlquotienten (rechts) in Abhängigkeit von n unter Gruppengröße 3 und probabilistisch wiederholter zyklischer schneller lokaler Suche

in Abbildung A.7 relativ zur schwarzen Glocke liegt, desto höhere Erfolgsquoten sind zu erwarten. Im Gegensatz zu der konstanten Länge weisen die Misserfolgstäler unterschiedliche Tiefen auf. Das zu $b = 1$ gehörende Tal ist am tiefsten. Die Wahrscheinlichkeit, dass $3b$ zufällige Läufe zum globalen Minimum ($\min f$) führen, ist monoton wachsend

in b . Bei sehr kleinen Parametern n und hinreichend großem b tritt oft der Fall ein, dass $\min_{K_0} = \min f = \min_{zf}$ gilt. Die Erfolgsquote liegt dann nahe bei 0.5. Dies ist eine mögliche Erklärung dafür, dass große Blockzahlen b für flache Misserfolgstäler sorgen.

In Abbildung A.9 links sind die durchschnittlichen g -Werte, d.h. Zielfunktionswerte nach Kriterium (1), der lokal minimalen Lösungen aus den Versuchen zu Blockzahl $b = 1$ dargestellt. Bei jedem Problemparameter $n \in \{1, 2, \dots, 60\}$ wurde über die g -Werte der 70 000 lokal minimalen Lösungen aus den zufälligen Läufen bzw. der 10 000 lokal minimalen Lösungen aus den Konsensläufen gemittelt. Man sieht, dass Konsensläufe im Durchschnitt bessere Funktionswerte liefern als zufällige Läufe. Schließlich sind in der Graphik rechts die **Schrittzahlquotienten** skizziert. Diese ergeben sich bei festem n als Quotient der durchschnittlichen Schrittzahl der 10 000 Konsensläufe und der durchschnittlichen Schrittzahl der 70 000 zufälligen Läufe. Für jedes $n \leq 60$ ist der Schrittzahlquotient kleiner als 1. Vermutlich konvergieren die Quotienten für $n \rightarrow \infty$ mit einem Grenzwert von höchstens 1.

A 2.3 Optimale Populationsgröße und Elternzahl im Genetischen Algorithmus mit wiederholter Mittelwertbildung

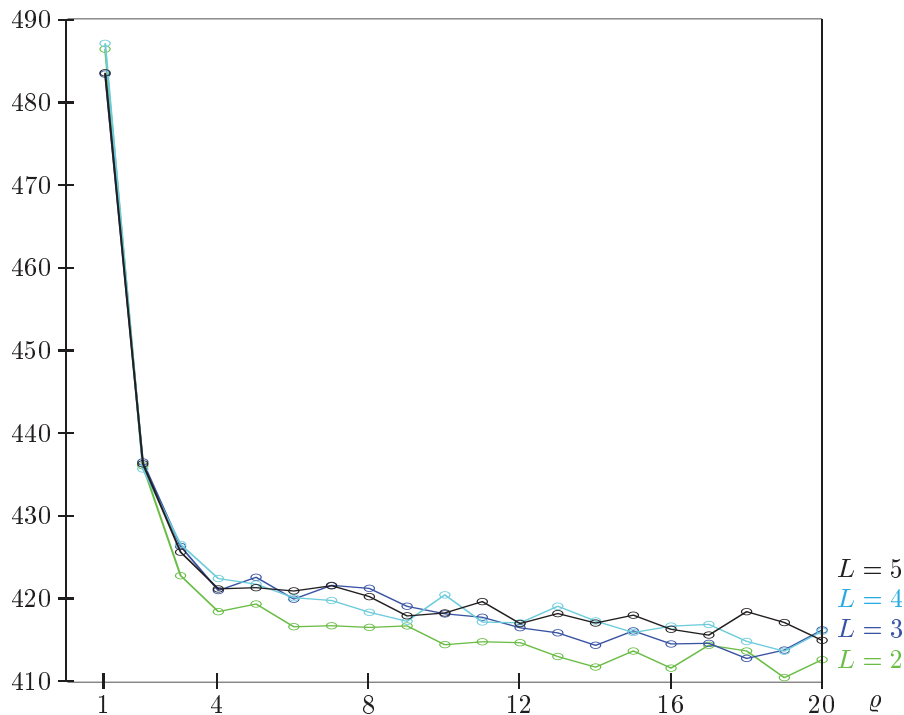


Abbildung A.10: **(Scheduling-Probleme)** Durchschnittlich erzielte kleinste g -Werte in Abhängigkeit von Populationsgröße ρ und Elternzahl L unter $n = 20$, $S = 99$ und probabilistisch wiederholter zyklischer schneller lokaler Suche bei Mittelwertbildung

In Abschnitt 4.2.2 wurde anhand des Shiftregister-Problems sowie des Dynamischen Optimierungsproblems beobachtet, dass Elternzahl 2 im Genetischen Algorithmus mit Mehrheitsbildung als Rekombinationsoperator optimal ist. Es wird nun die Frage untersucht,

ob dies auch für die Mittelwertbildung bei zufälligen Varianten von „Whizzkids 97“ in Algorithmus A.6 aus Abschnitt A 1.2 zutrifft.

Es war stets $S = 99$, d.h. es wurde jeweils 100 Generationen lang gerechnet. Ferner war die Zahl L der Eltern für die Rekombination fest aus $\{2, 3, 4, 5\}$ gewählt. Dann wurden für jede Poolgröße $\varrho \in \{1, 2, \dots, 20\}$ genau 1000 Probleminstanzen mit $n = 20$ Elternpaaren zufällig und unabhängig erzeugt. Nach Wahl einer Instanz wurde Algorithmus A.6 mit den Parametern L , ϱ und $S = 99$ aufgerufen. Dieser machte insgesamt $100 \cdot \varrho$ Läufe probabilistisch wiederholter zyklischer schneller lokaler Suche. Am Ende wurde über die besten g -Werte der 1000 Instanzen gemittelt. In Abbildung A.10 sind die durchschnittlichen besten g -Werte in Abhängigkeit von ϱ für jedes feste L skizziert.

Man kann zweierlei Beobachtungen machen: Für die Elternzahl zur Rekombination ist tatsächlich 2 optimal. Die besten gefundenen g -Werte werden im Durchschnitt mit wachsender Populationsgröße ϱ kleiner.

Fazit, Ausblick, offene Fragen

Am Ende dieser Arbeit werden die wichtigsten Beobachtungen zur Mehrheitsbildung zusammengefasst. Außerdem werden offene Fragen, Vermutungen und Ansätze für zukünftige Untersuchungen diskutiert.

Die Philosophie der Mehrheitsbildung

Die Mehrheitsbildung gemäß Definition 1.1, S. 13, ist eine Metaheuristik. Sie entfaltet ihre Wirkung zusammen mit „guten“ Heuristiken wie lokale Suche. Im Wesentlichen kommt es darauf an, dass die Heuristik, aufgefasst als Black Box, bei Eingabe einer beliebigen Startlösung irgendeine überdurchschnittlich gute Lösung der gegebenen Optimierungsaufgabe liefert. Durch wiederholte Anwendung der Heuristik mit unabhängigen Startlösungen lassen sich dann verschiedene gute Lösungen gewinnen. Mit Hilfe der Mehrheitsbildung werden strukturelle Gemeinsamkeiten dieser Lösungen in einem Mehrheitsvektor festgehalten. Bei Eingabe des Mehrheitsvektors erreicht die Heuristik typischerweise bessere Lösungen als bei Eingabe einer zufälligen Startlösung.

Erfolg und Misserfolg der Mehrheitsbildung

Im Rahmen dieser Dissertation wurden für zufällig erzeugte Instanzen verschiedener Optimierungsprobleme Versuche durchgeführt, in denen der beste erzielte Zielfunktionswert aus $L + 1$ zufälligen Läufen lokaler Suche mit dem besten erzielten Wert aus L zufälligen Läufen und einem anschließenden Mehrheitslauf verglichen wurde. Die Ergebnisse der Vergleiche wurden als empirische Erfolgsquoten zusammengefasst (Definition 2.1, S. 37). Die Erfolgsquote bildet einen Näherungswert für den Erfolg von Mehrheitsbildung (Definition 2.2, S. 38). Aus den Beobachtungen für die Erfolgsquoten in Abschnitt 2.2 ließ sich folgende Vermutung für den Erfolg von Mehrheitsbildung ableiten (Vermutungen 2.4, S. 40; 2.5, S. 53; 2.6, S. 56; 2.7, S. 61):

VERMUTUNG A.8

Der Erfolg von Mehrheitsbildung für

- *das Problem Max-3-SAT*
- *das Dynamische Optimierungsproblem*
- *das Shiftregister-Problem*
- *das Ising-Spinglas-Problem*

in Abhängigkeit des jeweiligen Problemparameters n konvergiert unter jeder Gruppengröße L und beliebiger in Abschnitt 1.1 genannter Variante wiederholter lokaler Suche für $n \rightarrow \infty$ gegen 1.

Der Erfolg von Mehrheitsbildung ist in folgender praktischer Situation von Bedeutung:

Angenommen, es ist eine Instanz für ein in Vermutung A.8 genanntes Optimierungsproblem mit Problemparameter n gegeben. Über eine spezielle Struktur der Instanz sei nichts bekannt. Zudem seien bereits L zufällige Läufe lokaler Suche gemacht worden. Die zur Verfügung stehende Rechenzeit erlaube nicht mehr als einen weiteren Lauf. Auf welche Weise soll die Startlösung für den abschließenden Lauf lokaler Suche gebildet werden?

Ist der Erfolg von Mehrheitsbildung unter n und L größer als $\frac{1}{2}$, ist statt einer zufällig gewählten Lösung ein Mehrheitsvektor als Startlösung ratsam. Nach Vermutung A.8 sind Mehrheitsläufe bei allen Problemparametern n bis auf endlich viele kleine n empfehlenswert. Kleine Problemparameter n liegen dagegen in einem Misserfolgstal. Misserfolgstäler wurden in Abschnitt 2.2 bei allen untersuchten Optimierungsproblemen beobachtet.

VERMUTUNG A.9

Der Erfolg von Mehrheitsbildung für

- *das Problem Max-3-SAT*
- *das Dynamische Optimierungsproblem*
- *das Shiftregister-Problem*
- *das Ising-Spinglas-Problem*
- *das Rucksack-Problem*

ist in einem Intervall kleiner Problemparameter n unter jeder Gruppengröße L und beliebiger in Abschnitt 1.1 genannter Variante wiederholter lokaler Suche höchstens $\frac{1}{2}$. Unter jeder Gruppengröße L existiert also ein Misserfolgstal.

Auch beim Counting-Ones-Problem wurden Misserfolgstäler beobachtet (Abschnitt 2.2.6). Hier wurden die Ausgangslösungen für die Bildung des Mehrheitsvektors nicht durch lokale Suche erzeugt, sondern unabhängig in jeder ihrer Komponenten durch Wahl von 1 mit Wahrscheinlichkeit $p > \frac{1}{2}$ und von 0 mit Wahrscheinlichkeit $1 - p$ (Definition 2.10 des p -Erfolgs, S. 65). Ferner wurde bewiesen, dass der p -Erfolg von Mehrheitsbildung für $n \rightarrow \infty$ gegen 1 konvergiert (Satz 2.11, S. 65). Lokale Suche ist also weder für die Existenz des Misserfolgstals noch die Konvergenz des Erfolgs gegen 1 notwendig. Dies gibt Anlass zu

VERMUTUNG A.10

Die Aussagen der Vermutungen A.8 und A.9 gelten nicht nur unter den in Abschnitt 1.1 genannten Varianten wiederholter lokaler Suche, sondern unter einer größeren Klasse von Verfahren, welche suboptimale Lösungen mit überdurchschnittlichem Zielfunktionswert liefern.

Die Frage nach einer genauen Charakterisierung dieser Klasse von Verfahren bleibt an dieser Stelle offen. Es erscheint jedoch als notwendig zu fordern, dass das Verfahren die Eingabe einer Startlösung erlaubt und die Ausgabelösung durch lokale Änderungen dieser Startlösung entsteht. Würde das Verfahren ungeachtet seiner Startlösung immer in derselben Lösung terminieren, wäre der Erfolg der Mehrheitsbildung trivialerweise $\frac{1}{2}$. Dies würde auch für ein Verfahren zutreffen, das sicher eine *global* optimale Lösung findet. Die Einschränkung auf *suboptimale* Lösungen, also Lösungen mit schlechterem Zielfunktionswert als dem globalen Optimum, ist in Vermutung A.10 notwendig.

Vermutlich gehören der Klasse von Verfahren in A.10 auch moderne Heuristiken wie Simulated Annealing ([26],[9]) oder Tabusuche ([18],[19]) an. Diese Heuristiken erfordern allerdings geeignet gewählte Parameter wie etwa bestimmte Wahrscheinlichkeitsverteilungen auf dem Lösungsraum. Die algorithmischen Parameter müssten für jedes Optimierungsproblem eigens optimiert werden. Es ist keineswegs klar, wie optimale Parameter für das Problem mit optimalen Parametern für die Mehrheitsbildung abzustimmen sind. Hier öffnet sich ein großes Experimentierfeld. Um Seiteneffekte zu vermeiden, wurde für die vorliegende Dissertation auf Experimente mit diesen Heuristiken verzichtet.

Erfolg und Misserfolg von randomisierten Mehrheitsbildungen

Analog zur Mehrheitsbildung wurden in Kapitel III dieser Dissertation weitere Bildungsregeln für Startlösungen lokaler Suche untersucht. Insbesondere wurden Vermutungen für den Erfolg verschiedener (p_0, p_1) -Regeln (Definition 3.4, S. 80) formuliert (Vermutung 3.3, S. 78). Zusammenfassend führen die Beobachtungen in Kapitel III zu folgender

VERMUTUNG A.11

Für unendlich viele $(p_0, p_1) \in [0, 1]^2$ konvergiert der Erfolg der (p_0, p_1) -Regel für

- das Problem Max-3-SAT
- das Dynamische Optimierungsproblem
- das Shiftregister-Problem
- das Ising-Spinglas-Problem

in Abhängigkeit des jeweiligen Problemparameters n unter beliebiger in Abschnitt 1.1 genannter Variante wiederholter lokaler Suche für $n \rightarrow \infty$ gegen 1. Diese Aussage gilt insbesondere für die $(0, 0)$ -Regel (= Mehrheitsregel), die $(0, \frac{1}{3})$ -Regel (= Proporzregel) sowie die $(0, \frac{1}{2})$ -Regel (= Einstimmigkeitsregel).

Ferner wurden auch für verschiedene (p_0, p_1) -Regeln Misserfolgstäter beobachtet:

VERMUTUNG A.12

Für unendlich viele $(p_0, p_1) \in [0, 1]^2$ ist der Erfolg der (p_0, p_1) -Regel

- beim Problem Max-3-SAT
- beim Dynamischen Optimierungsproblem
- beim Shiftregister-Problem
- beim Ising-Spinglas-Problem
- beim Rucksack-Problem

jeweils in einem Intervall kleiner Problemparameter n unter beliebiger in Abschnitt 1.1 genannter Variante wiederholter lokaler Suche höchstens $\frac{1}{2}$. Es existieren also Misserfolgstäter. Diese Aussage gilt insbesondere für Mehrheitsregel, Proporzregel und Einstimmigkeitsregel.

Analog zu A.10 lässt sich weiter vermuten:

VERMUTUNG A.13

Die Aussagen der Vermutungen A.11 und A.12 gelten nicht nur unter den in Abschnitt 1.1 genannten Varianten wiederholter lokaler Suche, sondern unter einer größeren Klasse von Verfahren, welche suboptimale Lösungen mit überdurchschnittlichem Zielfunktionswert liefern.

Faustregeln für randomisierte Mehrheitsbildungen

Angenommen, es ist eine Instanz für ein in Vermutung A.11 genanntes Optimierungsproblem mit Problemparameter n gegeben. Über eine spezielle Struktur der Instanz sei nichts bekannt. Zudem seien bereits drei zufällige Läufe lokaler Suche einer bestimmten Variante \mathcal{S} gemacht worden. Die zur Verfügung stehende Rechenzeit erlaube nicht mehr als einen weiteren Lauf. Nach welcher (p_0, p_1) -Regel soll die Startlösung für den abschließenden Lauf lokaler Suche gebildet werden?

Für Instanzen zu Max-3-SAT mit n Variablen lassen sich aus den Ergebnissen in Abschnitt 3.2 folgende Faustregeln ableiten:

FAUSTREGEL A.14

Liegt n außerhalb des Misserfolgstals von Mehrheitsbildung unter lokaler Suche der Variante \mathcal{S} , so ist die Mehrheitsregel, d.h. die $(0, 0)$ -Regel, empfehlenswert.

Gehört n zu den kleinen Parametern innerhalb des Misserfolgstals, so ist die Antimehrheitsregel, d.h. die $(1, 1)$ -Regel, empfehlenswert.

Gehört n zu den großen Parametern innerhalb des Misserfolgstals, so ist die $(\frac{1}{2}, \frac{1}{2})$ -Regel empfehlenswert. Hier ist also ein weiterer zufälliger Lauf ratsam.

Für Instanzen zum Ising-Spinglas-Problem auf dem $(n \times n)$ -Torus ohne äußeres Magnetfeld ergeben sich aus Abschnitt 3.2 folgende Faustregeln:

FAUSTREGEL A.15

Liegt n außerhalb des Misserfolgstals von Mehrheitsbildung unter lokaler Suche der Variante \mathcal{S} , so ist die Mehrheitsregel empfehlenswert.

Liegt n innerhalb des Misserfolgstals, so ist ein weiterer zufälliger Lauf lokaler Suche empfehlenswert.

Beim Shiftregister-Problem beschränkten sich die Versuche auf Mehrheitsregel, Proporzregel und Einstimmigkeitsregel. Die Antimehrheitsregel führt aufgrund von Symmetrieeigenschaften der Zielfunktion zu denselben Erfolgen wie die Mehrheitsregel. Auf ihren Einsatz wurde daher verzichtet. Für das Dynamische Optimierungsproblem wurden keinerlei Versuche mit (p_0, p_1) -Regeln außer der Mehrheitsregel durchgeführt. Trotzdem gelten für beide Probleme vermutlich zumindest die gleichen Faustregeln wie für das Ising-Spanglas-Problem.

Rekombination mit randomisierten Mehrheitsbildungen

Mehrheitsregel, Proporzregel und Einstimmigkeitsregel wurden in Kapitel IV zur Rekombination bei Genetischer Suche eingesetzt (Algorithmus 4.2, S. 93). Für die Mehrheitsregel wurden bei etlichen Benchmark-Instanzen zu Max-SAT, dem Shiftregister-Problem und dem Ising-Spینگlas-Problem folgende Beobachtungen gemacht: Die gefundenen lokalen Optima tendieren dazu, mit wachsender Generationenzahl besser zu werden. Die Schrittzahlen werden dabei tendenziell kleiner. Beide Effekte treten ohne Zusatzmaßnahmen wie Elitismus, Mutation oder fitnessabhängige Selektion auf (Abschnitt 4.2.1). Für zufällig gewählte Instanzen des Shiftregister-Problems wurden ferner Untersuchungen zur Optimierung von Populationsgröße und Elternzahl gemacht. Aufgrund der Beobachtungen lassen sich folgende Vermutungen formulieren (Abbildungen 4.8, S. 98; 4.10, S. 100; 4.12, S. 102):

VERMUTUNG A.16

Betrachte das Shiftregister-Problem. Im Genetischen Algorithmus 4.2 seien die Populationsgröße $q \geq 1$ sowie eine hinreichend große Generationenzahl fest gewählt. Dann gilt im Durchschnitt über alle Instanzen zum Parameter 1000:

1. *Wird zur Rekombination die Mehrheitsbildung oder die Proporzbildung eingesetzt, so ist Elternzahl 2 optimal.*
2. *Wird zur Rekombination die Einstimmigkeitsbildung eingesetzt, so ist Elternzahl 3 optimal, falls q hinreichend groß ist. Bei kleinen Populationsgrößen q liefern große Elternzahlen bessere f -Werte als kleine Elternzahlen.*

Das Optimalitätskriterium ist dabei jeweils der beste Zielfunktionswert, der in den Populationen aller Generationen insgesamt auftritt.

Liegt eine Instanz für das Shiftregister-Problem vor, von der keine strukturellen Eigenschaften bekannt sind, so lässt sich folgende Empfehlung ableiten:

FAUSTREGEL A.17

Im Genetischen Algorithmus 4.2 ist zur Rekombination die Einstimmigkeitsregel ratsam. Aus Effizienzgründen sind kleine Populationsgrößen empfehlenswert. Neben Elternzahl 2 sollten auch größere Elternzahlen probiert werden.

Diese Faustregel kann auch für das Dynamische Optimierungsproblem übernommen werden (Abbildungen 4.9, S. 99; 4.11, S. 101; 4.13, S. 103).

Offene Fragen

Es liegt in der Natur der Sache, dass Computer-Experimente nur endlich viele Spezialfälle von allgemein formulierten Fragestellungen erfassen können. Aus jeder Beobachtung resultieren etliche neue offene Fragen. Im Folgenden wird eine kleine Auswahl offener Fragen zur Thematik der Mehrheitsbildung mit Ansätzen für weitere Untersuchungen aufgelistet:

1. Wie lassen sich die Beobachtungen bzw. Vermutungen dieser Arbeit mathematisch beweisen? Wie allgemeingültig sind die Aussagen?
2. Welche Eigenschaften sind für ein Optimierungsproblem notwendig bzw. hinreichend, damit der Erfolg einer bestimmten (p_0, p_1) -Regel gegen 1 konvergiert?
3. Wie lassen sich die Misserfolgstäler beweisen? Wie gelingt ein Beweis beim einfachen Counting-Ones-Problem? Gibt es Optimierungsprobleme ohne Misserfolgstäler?
4. Gibt es gewisse Problemtransformationen, unter denen die Konvergenzeigenschaften des Erfolgs invariant sind?
5. Wie lassen sich die in den Vermutungen A.10 und A.13 genannten Klassen von Verfahren charakterisieren? Konvergiert der Erfolg von (randomisierter) Mehrheitsbildung einer besten bekannten Heuristik für ein gegebenes Optimierungsproblem gegen 1? Inwieweit wirkt sich die Güte der Heuristik auf die Konvergenzgeschwindigkeit aus?
6. Welche Beobachtungen lassen sich machen, wenn reale Laufzeiten untersucht werden? Wie effizient sind (randomisierte) Mehrheitsläufe gegenüber zufälligen Läufen lokaler Suche?
7. Sei eine reale Laufzeitschranke vorgegeben. Es seien entweder nur zufällige Läufe wiederholter lokaler Suche oder eine bestimmte Kombination zwischen zufälligen Läufen und Mehrheitsläufen erlaubt. Bei beiden Varianten darf so lange gerechnet werden, bis das Zeitlimit erreicht wird. Existieren Misserfolgstäler, falls der „Erfolg“ der Mehrheitsbildung durch Vergleichen des jeweils besten gefundenen Zielfunktionswerts bei der Varianten gemessen wird?
8. Bei Max-3-SAT wurde beobachtet, dass die Mehrheitsbildung besonders erfolgreich ist, wenn die Probleminstanzen in gewissem Sinne schwierig sind (Abschnitt 2.2.1, S. 51). Lassen sich auch bei anderen Optimierungsproblemen ähnliche Feststellungen treffen? Beispielsweise könnte man die Schwierigkeit einer Ising-Spinglas-Instanz als die Zahl der Kreise der Länge 4 auf dem $(n \times n)$ -Torus messen, deren Interaktionen nicht sämtlich erfüllbar sind.
9. Welche Möglichkeiten bestehen, die Idee der Mehrheitsbildung im Lösungsraum $\{0, 1\}^n$ als eine Art Konsensbildung auf andere Lösungsräume zu übertragen? Ein Beispiel hierfür ist die Mittelwertbildung bei Scheduling-Problemen (Definition A.2, S. 107). Welche Verallgemeinerungen sind bei Produkträumen $V = V_1 \times \dots \times V_n$ sinnvoll, wenn $\#V_i > 2$ für mindestens ein i gilt?
10. Ist beim Rucksack-Problem die Konvergenz des Erfolgs von Mehrheitsbildung gegen 1 möglich, wenn andere Zielfunktionen als die in Abschnitt 1.2.5 definierten Funktionen $f : \{0, 1\}^n \rightarrow \mathbb{R}$ verwendet werden? Ist das sublineare Wachstum von $\bar{f}_{\text{Mh}}(n) - \bar{f}_{\text{zf}}(n)$ mit n (vgl. S. 62) für den Misserfolg der Mehrheitsbildung notwendig?
11. Wie gut ist Genetische Suche mit (randomisierter) Mehrheitsbildung, wenn Zusatzmechanismen wie Elitismus, Mutation oder fitnessabhängige Selektion angewendet werden?

Symbolverzeichnis

\emptyset	leere Menge
$\#A$	Mächtigkeit der Menge A
\neg	Negation einer BOOLEschen Variablen
\vee	logisches Oder
\wedge	logisches Und
\sim	Äquivalenzrelation
2^Ω	Potenzmenge von Ω
$\{i : E\}$	Menge aller Elemente i mit Eigenschaft E ($= \{i \mid E\}$)
$\{i \mid E\}$	Menge aller Elemente i mit Eigenschaft E ($= \{i : E\}$)
$\{0, 1\}^n$	Menge aller Bitstrings der Länge n
$\{-1, +1\}^T$	Menge aller Abbildungen $T \rightarrow \{-1, +1\}$
$\lfloor x \rfloor$	$\max \{y \in \mathbb{Z} \mid y \leq x\}$
$\lceil x \rceil$	$\min \{y \in \mathbb{Z} \mid y \geq x\}$
$[x]$	Äquivalenzklasse von x
$[a, b]$	$\{c \mid a \leq c \leq b\}$ abgeschlossenes Intervall von a bis b
$]a, b[$	$\{c \mid a < c < b\}$ offenes Intervall von a bis b
$[a, b[$	$\{c \mid a \leq c < b\}$ halboffenes Intervall von a bis b
$]a, b]$	$\{c \mid a < c \leq b\}$ halboffenes Intervall von a bis b
$a \bmod b$	Rest bei Division von a durch b
B	Beispielzahl
d_H	HAMMING-Abstand
$E[X]$	Erwartungswert der reellen Zufallsvariable X
f	Zielfunktion des Optimierungsproblems
$G_{v,s}$	Elternindexmengen zur Rekombination im Genetischen Algorithmus
i.i.d.	independent identically distributed, unabhängig und identisch verteilt
L	Anzahl an Läufen, Gruppengröße, Elternzahl
$m^* = m^*(n)$	$\lceil 4.25 n \rceil$
$\max A$	größtes Element von A
$\max f$	größtes Element der Wertemenge der Funktion f
$\min A$	kleinstes Element von A
$\min f$	kleinstes Element der Wertemenge der Funktion f
$n^*(m)$	$\lfloor (4.25)^{-1} m \rfloor$
\mathbb{N}_0	$\{0, 1, 2, \dots\}$ Menge der natürlichen Zahlen einschließlich 0
\mathbb{N}^+	$\{1, 2, 3, \dots\}$ Menge der natürlichen Zahlen ausschließlich 0
$O(n)$	$\{g : \mathbb{N}^+ \rightarrow \mathbb{N}^+ : \exists c > 0 \exists n_0 \in \mathbb{N}^+ \forall n \geq n_0 : g(n) \leq c \cdot n\}$
Ω	Wahrscheinlichkeitsraum
P, P_i	Wahrscheinlichkeitsmaße
\mathcal{P}	Pool, Population im Genetischen Algorithmus
p_0, p_1, p_2, p_3	Wahrscheinlichkeiten für randomisierte Mehrheitsbildungen
π	Permutation
φ^{-1}	Umkehrfunktion von φ
\mathbb{R}	Menge der reellen Zahlen
ϱ	Poolgröße, Populationsgröße im Genetischen Algorithmus

S	maximale Generation im Genetischen Algorithmus
S_n	Menge der Permutationen von $\{1, 2, \dots, n\}$
$(S_v)_{v \in T}$	durch T indizierte Familie
T	Termin, Task
V	Lösungsraum des Optimierungsproblems
$V[X]$	Varianz der reellen Zufallsvariable X
$y^i(x)$	i -ter HAMMING-Nachbar des Vektors $x \in \{0, 1\}^n$, d.h. Element von $\{0, 1\}^n$, das durch Flippen der i -ten Komponente aus x entsteht
$y^{ij}(x)$	Element von $\{0, 1\}^n$, das durch Flippen der i -ten sowie der j -ten Komponente aus $x \in \{0, 1\}^n$ entsteht ($1 \leq i < j \leq n$)
\mathbb{Z}	Menge der ganzen Zahlen
$\mathbb{Z}/k\mathbb{Z}$	Restklassenring von \mathbb{Z} modulo $k\mathbb{Z}$

Literaturverzeichnis

- [1] AARTS, E.H.L., LENSTRA, J.K. (ed.), *Local Search in Combinatorial Optimization*, John Wiley & Sons, Chichester 1997.
- [2] ALTHÖFER, I., *Je mehr, desto besser? Mehrheitssysteme im Computerschach und der Yazgac-Test*, MODUL (Juni 1990), 54-61.
- [3] ALTHÖFER, I., *Selective trees and majority systems: two experiments with commercial chess computers*, in: BEAL, D.F. (ed.), *Advances in Computer Chess 6*, Ellis Horwood, Chichester 1991, 37-59.
- [4] ARLT, H.-D., *Gewichtete Mehrheitssysteme am Beispiel des Computerschachs*, Diplomarbeit, Fakultät für Mathematik und Informatik, Friedrich-Schiller-Universität Jena, Jena 2000.
- [5] BARAHONA, F., *On the computational complexity of Ising spin glass models*, Journal of Physics A **15** (1982), 3241-3253.
- [6] BARAHONA, F., *Ground-state magnetization of Ising spin glasses*, Physical Review B **49** (1994), 12864-12867.
- [7] BELLMANN, R., *Dynamic Programming*, Princeton University Press, Princeton 1957.
- [8] BRUCKER, P., *Scheduling Algorithms*, 2. Auflage, Springer, Berlin 1998.
- [9] ČERNÝ, V., *Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm*, Journal of Optimization Theory and Applications **45** (1985), 41-51.
- [10] COOK, S.A., *The complexity of theorem-proving procedures*, Proceedings 3rd Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, New York 1971, 151-158.
- [11] DAVIS, M., PUTNAM, H., *A computing procedure for quantification theory*, Journal of the Association for Computing Machinery **7** (1960), 201-215.
- [12] DIEHL, M., JÜNGER, M., MUTZEL, P., REINELT, G., RINALDI, G., DE SIMONE, C., *Exact ground states of Ising spin glasses: New experimental results with a branch and cut algorithm*, Journal of Statistical Physics **80** (1995), 487-496.
- [13] DIEHL, M., JÜNGER, M., MUTZEL, P., REINELT, G., RINALDI, G., DE SIMONE, C., *Exact ground states of two-dimensional $\pm J$ Ising spin glasses*, Journal of Statistical Physics **84** (1996), 1363-1371.
- [14] EIBEN, A.E., RAUÉ, P.-E., RUTTKAY, Z., *Genetic algorithms with multi-parent recombination*, in: DAVIDOR, Y., SCHWEFEL, H.-P., MÄNNER, R. (ed.), *Parallel Problem Solving from Nature — PPSN III*, Lecture Notes in Computer Science **866**, Springer, Berlin 1994, 78-87.
- [15] GAREY, M.R., JOHNSON, D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York 1979.
- [16] GAREY, M.R., JOHNSON, D.S., SETHI, R., *The complexity of flowshop and jobshop scheduling*, Mathematical Operations Research **1** (1976), 117-129.

- [17] GAREY, M.R., JOHNSON, D.S., STOCKMEYER, L., *Some simplified NP-complete graph problems*, Theoretical Computer Science **1** (1976), 237-267.
- [18] GLOVER, F., *Tabu search: part I*, ORSA Journal on Computing **1** (1989), 190-206.
- [19] GLOVER, F., *Tabu search: part II*, ORSA Journal on Computing **2** (1990), 4-32.
- [20] GOLDBERG, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA (1989).
- [21] GOLOMB, S.W., *Shift Register Sequences (2nd ed.)*, Aegean Park Press, Laguna Hills 1982.
- [22] HOLLAND, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI (1975).
- [23] JOHNSON, D.S., TRICK, M.A. (ed.), *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, October 11-13, 1993*, AMS, Providence, RI 1996.
- [24] JÜNGER, M., RINALDI, G., *Relaxations of the Max Cut Problem and computation of spin glass ground states*, in: KISCHKA, P., LORENZ, H.W., DERIGS, U., DOMSCHKE, W., KLEINSCHMIDT, P., MÖHRING, R. (ed.), *Operations Research Proceedings 1997, Selected Papers of the Symposium on Operations Research (SOR '97)*, Springer, Berlin 1998, 74-83.
- [25] KARP, R.M., *Reducibility among combinatorial problems*, in: MILLER, R.E., THATCHER, J.W. (ed.), *Complexity of Computer Computations*, Plenum Press, New York 1972, 85-103.
- [26] KIRKPATRICK, C.D., GELATT, JR., VECCHI, M.P., *Optimization by simulated annealing*, Science **220** (1983), 671-680.
- [27] LEVESQUE, H., MITCHELL, D., SELMAN, B., *Hard and easy distributions of SAT problems*, Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92), San Jose, CA, July 1992, 459-465.
- [28] LIEBSCHER, V., WALCHER, S., *On the mean value of probability measures on circular graphs*, preprint, Institut für Biomathematik und Biometrie, GSF-Forschungszentrum, Neuherberg 1999.
- [29] NISSEN, V., *Einführung in Evolutionäre Algorithmen: Optimierung nach dem Vorbild der Evolution*, Vieweg, Wiesbaden 1997.
- [30] SCHOOF, J., *Kooperative Optimierung mit kommunizierenden Genetischen Algorithmen*, Dissertation, Fakultät für Mathematik und Informatik, Julius-Maximilians-Universität Würzburg, Würzburg 1998.

Index

- AARTS, 104, 105, 131
 Äquivalenzrelation
 für Shiftregisterlösungen, 54
 für Spinzustände, 58, 78
 Algorithmus
 Genetischer, 90–103, 113–116, 121–122
 ALTHÖFER, 6, 7, 131
 Anti-Mehrheit, 24
 Antimehrheitsbildung, **73**
 Antimehrheitslauf, 9, **73**
 Antimehrheitsregel, 9, **73**, 126
 Antimehrheitsvektor, **73**
 Approximationsalgorithmus, 12
 ARLT, 7, 131
 Austausch-Nachbarschaft, **107**

 BARAHONA, 26, 131
 BELLMANN, 23, 131
 Bisexuelle Rekombination, 99
 Bitfeldmethode, 19
 Black Box, 7, 123
 BOOLE, 21
 BOOLEsche Formel, 21
 BOOLEsche Funktion, 35
 BOOLEsche Variable, 21
 Branch and Bound, 105
 Branch and Cut, 27
 BRUCKER, 131

 ČERNÝ, 131
 CHVATAL, 30
 Computerschach, 6
 COOK, 21, 30, 131
 Counting-Ones-Problem, 14, **29**, 63–66
 Couting-Ones-Problem, 124

 DAVIDOR, 131
 DAVIS, 39, 131
 Defekt, **24**
 DERIGS, 132
 DE SIMONE, 131
 De Telegraaf, 104
 Deterministische schnelle lokale Suche, **18**
 Deterministische volle lokale Suche, **18**
 Deterministisch wiederholte lokale Suche, **18**
 DIEHL, 131
 DIMACS, 30
 Diversität, 95–99, 101

 DOMSCHKE, 132
 Doppel-Flip-Nachbarschaft, **29**, 61
 Drehen
 einer Shiftregister-Lösung, 54, 103
 Dynamische Optimierung, 23
 Dynamisches Optimierungsproblem, **22**, 52–
 53, 68–71, 78, 99, 101, 103, 119, 121,
 123

 Echtzeit-Bedingungen, 9
 EIBEN, 131
 Einstimmigkeitsbildung, 9, **73**, 101–103
 Einstimmigkeitslauf, **73**
 Einstimmigkeitsregel, **72**, 100, 125
 Einstimmigkeitsvektor, **73**
 Einstufige Mehrheitsbildung, 30–85
 Elitismus, 114
 Elternwahl, 94, 115
 Elternzahl, 94
 Energie
 eines Spinzustands, **25**
 Erfolg, **38**
 der (p_0, p_1) -Regel, **80**
 von Antimehrheitsbildung, **73**
 von Einstimmigkeitsbildung, **73**
 von Mehrheitsbildung, **38**, 123
 von Mittelwertbildung, **119**
 von Proporzbildung, **73**
 Erfolgshügel, 73
 Erfolgskurve, **38**
 Pool-, **70**
 von Antimehrheitsbildung, **73**
 von Einstimmigkeitsbildung, **73**
 von Mehrheitsbildung, **38**
 von Proporzbildung, **73**
 Erfolgsquote, **37**
 der (p_0, p_1) -Regel, **80**
 Pool-, **70**
 von Antimehrheitsbildung, **73**
 von Einstimmigkeitsbildung, **73**
 von Mehrheitsbildung, **37**, 123
 von Mittelwertbildung, **119**
 von Proporzbildung, **73**
 Evolution, 93

 Färbbarkeit von Graphen, 35
 Fehleranalyse von Schaltkreisen, 35

- Fitness, 94, 115
 Flippen
 einer Lösungskomponente, 18
 einer Shiftregister-Lösung, 54, 103
 eines Spinzustands, 59, 77
 von Lösungskomponenten (Mutation), 97
 zweier Lösungskomponenten, 29
 Fragenzahl, **16**

 GANTT-Diagramm, 109
 GAREY, 22, 105, 131, 132
 GAUSS-Verteilung, 27
 GELATT, 132
 Gene Scanning, 100
 Genetische lokale Suche, 97
 Genetischer Algorithmus, 90–103, 113–116,
 121–122
 hybrider, 97
 Genetische Suche, 9, **97**
 GLOVER, 132
 GOLDBERG, 94, 132
 GOLOMB, 132
 Graphen
 Färbbarkeit von, 35
 Grundzustand, 25
 Gruppengröße, **13**

 HAMMING-Abstand, **16**, 34, 54, 55, 67
 HAMMING-Nachbar, **14**
 HAMMING-Nachbarschaft, **14**
 beim ± 1 -Ising-Spینگlas-Problem, **27**
 Hanoi
 Türme von, 30, 35
 Hanoi-5, 30–34, 67, 70, 86–90, 94–97
 Heuristik, 6, 66, 123
 HOLLAND, 93, 132
 HOGENDIJK, 105
 Hybrider Genetischer Algorithmus, 97

 Individuum, 93
 Interaktion, 25
 Erfüllung einer, 25
 Ising-Spینگlas, 25
 Ising-Spینگlas-Problem, **25**, 58–61, 75–78, 80,
 84–85, 97, 123

 $\pm J$ -Ising-Modell, 25
 $\pm J$ -Modell, 25
 Job-Shop Scheduling, 105
 JOHNSON, 22, 30, 105, 131, 132

 JÜNGER, 131, 132

 Kanonische Reihenfolge
 der Nachbarn bzgl. HAMMING-N., 15, 17,
 31
 der Nachbarn bzgl. Doppel-Flip-N., 61
 der Nachbarn bzgl. P.-Austausch-N., 107
 KARP, 28, 132
 KIRKPATRICK, 132
 KISCHKA, 132
 Klausel, **21**, 38
 gemischte, **49**
 Länge einer, 22
 reine, **49**, 75
 KLEINSCHMIDT, 132
 KNF, 21
 Kodierungstheorie, 23
 Kombinatorische Optimierung, 6, 11
 Konjunktive Normalform, 21
 Konsensbildung, 110
 Konsenslauf, 10, **111**
 Kryptographie, 23
 k -SAT, 22
 Kurve der Schrittzahlquotienten
 von Antimehrheitsbildung, **73**
 von Einstimmigkeitsbildung, **73**
 von Proporzbildung, **73**

 Lauf lokaler Suche, **14**
 Konsenslauf, 10, **111**
 Mehrheitslauf, 7, **14**, 30
 zufälliger, 7, **14**, 30, **111**
 LENSTRA, 104, 105, 131
 LEVESQUE, 39, 132
 LIEBSCHER, 7, 132
 Literal, **21**, 38
 Lösung, 6, **11**
 global maximale, 6, **11**
 global minimale, 6
 lokal maximale, **14**
 lokal minimale, **21**
 suboptimale, 6, 12, 124
 Lösungsraum, **11**
 Lokale Suche, 7, 14–21, 123
 Antimehrheitslauf von, **73**
 deterministische schnelle, **18**
 deterministische volle, **18**
 Einstimmigkeitlauf von, **73**
 Genetische, 97
 Konsenslauf von, 10, **111**

- Lauf von, **14**
 - Mehrheitslauf von, 7, **14**, 30, 73
 - probabilistische schnelle, **18**
 - probabilistische volle, **18**
 - Proporzlauf von, **73**
 - schnelle, **16**
 - deterministische, **18**
 - probabilistische, **18**
 - wiederholte, **18**
 - volle, **15**
 - deterministische, **18**
 - probabilistische, **18**
 - wiederholte, **14**
 - deterministisch, **18**
 - probabilistisch, **18**
 - volle, **18**
 - zufälliger Lauf von, 7, **14**, 30, **111**
 - zyklische schnelle, **17**
 - wiederholte, **18**
- LORENZ, 132
- MÄNNER, 131
- Magnetfeld
 - äußeres, 25
- Maschinelles Lernen, 94
- Max-3-SAT, 38–51, 73–75, 78, 80–84, 123
- Maximierungsaufgabe, 6, **11**
- Maximierungsproblem, 6
- Max- k -SAT, **22**, 80
- Max-SAT, 12, **22**
- MCGEOCH, 30
- Median, 7, **67**, 73
- Medianbildung, 7, 67
- Mehrheitsbildung, 6, **13**, 73, 123
 - einstufige, 30–85
 - mehrstufige, 86–90
 - randomisierte, 8, 72–85
 - allgemeines Modell, 80–85
- Mehrheitslauf, 7, **14**, 30, 73
 - erster Stufe, **86**
 - zweiter Stufe, **86**
- Mehrheitsregel, **72**, 125
- Mehrheitsvektor, 6, **13**, **73**, 123
- Mehrstufige Mehrheitsbildung, 86–90
- Meta-Heuristik, 66, 123
- MILLER, 132
- Minimierungsaufgabe, **21**
- Misserfolgstal, 8, 10, 40, **40**, 47, 53, 56, 61, 62, 66–71, 73, 75, 119, 124
- MITCHELL, 39, 132
- Mittelwertbildung, 10, **108**
- MÖHRING, 132
- Multisexuelle Rekombination, 99
- Mustererkennung, 94
- Mutation, 93
- MUTZEL, 131
- NISSEN, 132
- N Point Crossover, 100
- NP-schwer, 12, 27
- NP-vollständig, 6, 12, 21, 105
- Ökonomie, 94
- Optimierung
 - Dynamische, 23
 - Kombinatorische, 6, 11
- Optimierungsaufgabe, 6, 11
- (p_0, p_1) -Regel, 8, **80**, 125
- (p_0, p_1) -Vektor, **80**
- Paar-Austausch-Nachbarschaft, **107**
- Paket, 27
 - Größe eines, 27, 61
 - Wert eines, 27, 61
- Parameteridentifikation, 35
- p -Erfolg, **65**, 124
- p -Erfolgskurve, **64**
- p -Erfolgsquote, **63**
- Plateau, 19, 97
- Pool, 70, 91
- Pool-Erfolgskurve, **70**
- Pool-Erfolgsquote, **70**
- Poolgröße, 93
- Population, 93
- Populationsgröße, **94**
- Probabilistische schnelle lokale Suche, **18**
- Probabilistische volle lokale Suche, **18**
- Probabilistisch wiederholte lokale Suche, **18**
- Problemparameter, **36**
- Proporzbildung, **73**, 100–101
- Proporzlauf, **73**
- Proporzregel, **72**, 100, 125
- Proporzvektor, **73**
- Pseudo-Zufallszahlen-Generator, 23
- PUTNAM, 39, 131
- Randomisierte Mehrheitsbildungen, 8, 72–85
 - allgemeines Modell, 80–85
- Random Search, 12

- RAUÉ, 131
- REINELT, 131
- Rekombination, 93, 115, 122
 - bisexuelle, 99
 - multisexuelle, 99
- Rekombinationsoperator, 9, 94
- Resolutionsverfahren, 39
- RINALDI, 131, 132
- Rucksack, 27
- Rucksackgröße, 27, 61, 62
- Rucksackladung, 28
- Rucksack-Problem, **27**, 61–63, 78–79, 124
- Rückkopplungsfunktion, 23
- Rückwärtsrekursion, 23
- Rundenmethode, 19
- RUTTKAY, 131

- SAT, **21**
- Satisfiability-Problem, **21**
- Schaltkreise
 - Fehleranalyse von, 35
- Schedule
 - zulässiger, **106**
- Schnelle lokale Suche, **16**
 - deterministische, **18**
 - probabilistische, **18**
 - wiederholte, **18**
- SCHOOF, 132
- Schrankenmethode, **19**, 32, 96, 115
- Schrittzahl, **16**
- Schrittzahlquotienten, **38**
 - der (p_0, p_1) -Regel, **80**
 - Kurve der, **38**, 73
 - von Antimehrheitsbildung, **73**
 - von Einstimmigkeitsbildung, **73**
 - von Mehrheitsbildung, **38**
 - von Mittelwertbildung, **121**
 - von Proporzbildung, **73**
- SCHWEFEL, 131
- Selektion, 93, 115
- SELMAN, 39, 132
- SETHI, 105, 131
- Shiftregister, 23
- Shiftregisterfolge, 23
- Shiftregister-Problem, **24**, 53–57, 68, 75, 78,
 - 80, 97, 98, 100, 101, 121, 123
- Sidestep, **19**, 32–34, 86–88, 96, 115
- Simulated Annealing, 66, 125
- Spieltheorie, 94
- Spin, 25
- Spinzustand, **25**, 27
 - äquivalenter, 58
 - Energie des, **25**
- STOCKMEYER, 22, 132
- Strafterm, 28, 61

- Tabusuche, 66, 125
- TARJAN, 30
- Termin, **105**
- THATCHER, 132
- Torus, 26
- TRICK, 30, 132
- TSCHEBYSCHEFF
 - Ungleichung von, 65
- Türme von Hanoi, 30, 35

- Uniform Crossover, 100

- VECCHI, 132
- Volle lokale Suche, **15**
 - deterministische, **18**
 - probabilistische, **18**
 - wiederholte, **18**
- Vorwärtsrekursion, 23

- WALCHER, 7, 132
- Whizzkids-'97, 104, 111–116
- Whizzkids Mathematics Contest, 104
- Wiederholte lokale Suche, **14**
 - deterministisch, **18**
 - probabilistisch, **18**
- Wiederholte schnelle lokale Suche, **18**
- Wiederholte volle lokale Suche, **18**
- Wiederholte zyklische schnelle lokale Suche,
 - 18**

- Zeit-Einfüge-Prozedur, **106**, 108
- Zielfunktion, **11**
- Zufälliger Lauf, 7, **14**, 30, **111**
- Zulässiger Schedule, **106**
- Zyklische schnelle lokale Suche, **17**
 - wiederholte, **18**

Selbständigkeitserklärung

Ich erkläre, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Hilfsmittel und Literatur angefertigt habe.

Ort, Datum

Unterschrift des Verfassers

Lebenslauf

Rose, Claus

geb. 13.09.1968

Vilshofen

1975-1979

Grundschule Ritter Tuschl, Vilshofen

1979-1988

Gymnasium Vilshofen

1988

Abitur

1989-1996

Studium der Mathematik mit Nebenfach Informatik
an der Universität Passau

2.5.1996

Abschluss zum Dipl.-Math. an der Universität Passau

1996-2000

Wissenschaftlicher Mitarbeiter
am Institut für Angewandte Mathematik
der Friedrich-Schiller-Universität Jena

Ort, Datum

Unterschrift